UNIVERSITY OF BERGEN
DEPARTMENT OF MATHEMATICS

Modern Variable Selection Methods with Empirical Analysis

Author: Mathias Einarsen Ostnes

Supervisors: Yushu Li and Ingvild Margrethe Helgøy

June 1, 2023

Acknowledgements

I would like to express my deepest gratitude and appreciation to the following individuals and organizations who have played a significant role in the completion of my master's thesis:

My main supervisor, Yushu Li, and co-supervisor, Ingvild Helgøy, both deserve special mention for their unwavering support, guidance, and expertise throughout this research endeavor. Their valuable insights, constructive feedback, and encouragement have been instrumental in shaping the outcome of this thesis.

I would also like to extend my heartfelt thanks to the Department of Mathematics and especially the Statistics and the data science group at UiB for providing a conducive academic environment and necessary resources that facilitated the successful completion of this thesis. The department's commitment to excellence and dedication to fostering a culture of learning have been invaluable.

To my peers and fellow classmates, I am incredibly grateful for your friendship. It has been a pleasure sharing these academic years with you all.

Mathias Ostnes
Bergen, 01.June

Abstract

In the realm of modeling with big data including high-dimensional datasets, the challenge lies in extracting the most relevant and informative information while avoiding overfitting of general models, especially when it comes to prediction based on the given dataset. This thesis focuses on utilizing sparse methods especially sparse Bayesian learning methods to construct models that mitigate the risk of overfitting by utilizing only the most crucial aspects of the data in the framework of supervised learning. By employing these well-developed techniques, the most informative observations or variables can be extracted to reveal the systematic pattern of the dataset as well as further prediction. Six methods are examined, including well-known techniques such as LASSO, Ridge Regression, Bayesian Lasso, and the relevance vector machine (RVM), as well as two recently developed methods: RVM$_{BLS}$ and RVM$_{BLSX}$. The latter, RVM$_{BLSX}$ is proposed by the author of this thesis.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Motivation and Contribution

In the current day and age, methods for gathering data have become easier and more accessible all over the world. This leads to a vast amount of large datasets for data analysis with the aim of discovering systematic patterns and relationships within the datasets. The extracted or revealed knowledge from existing data can even help further predictions. The corresponding drawback for analyzing "big" datasets is that it will be more complicated and computationally expensive to implement mathematical and statistical learning models as well as utilizing computational techniques to extract information from data. Especially when it comes to the prediction using existing datasets. We can observe large datasets, whereas we only need the most informative data which can reflect the systematic structure of the data patterns, instead of using the whole dataset which can often include redundant information. This thesis will focus on ways of analyzing datasets that contain redundant information in supervised learning. Our main focus will be methods of Sparse Bayesian learning that can achieve variable selection and sample size reduction in the framework of Bayesian Analysis. Sparse Bayesian learning is one type of supervised learning where we try to extract the most informative parts of datasets for probabilistic prediction, whilst still achieve point estimation, when using the mean or the mode of the predictive distribution for the posterior distribution as point estimates for prediction. Our focus will be on the following six methods: Least Absolute Shrinkage and Selection Operator (LASSO), Ridge Regression, Bayesian Lasso, The relevance vector machine (RVM), RVM$_{BLS}$ and RVM$_{BLSX}$. While the first four are quite well known, the last two method are recently developed, whereas RVM$_{BLSX}$ is my modification for RVM$_{BLS}$ [Helgøy and Li, 2023]. These methods will be illustrated within the Bayesian framework and among them are Lasso, RVM, RVM$_{BLS}$ and RVM$_{BLSX}$ can achieve variable selection/dimensional reduction and sample selection/sample size reduction by automatic data driven algorithms, in the way that certain estimated weight parameters of variables or samples will be set to zero after the learning process, while Ridge and Bayesian Lasso can not. For Ridge and Bayesian Lasso we need to identify a manual threshold for the estimated weight coefficient if we want to achieve variable selection whereas the weights with estimated values under this threshold being set to zero. This thesis will present detailed mathematical modeling and statistical inference process of those methods, compare them, and implement those methods using different empirical datasets.

My own contribution in this thesis is the method tentatively named RVM_{BLSX}, which is an extension of RVM_{BLS} that [Helgøy and Li, 2023] developed. RVM_{BLS} is an extension of the RVM from [Tipping, 2001], as the RVM_{BLS} use the type-II maximum likelihood estimation method implemented in RVM to estimate the hyperparameter in prior, while the hierarchical structure in RVM_{BLS} is from the Bayesian Lasso model ([Park and Casella, 2008][Helgøy and Li, 2023]). In RVM_{BLS}, the hyperparameter in both hierarchical prior and hyperprior are estimated by maximizing the same marginal likelihood with the respect to the hyperparameters separately. RVM_{BLSX} utilize Gibbs sampling method in [Park and Casella, 2008] to identify the hyperparameter in the hyperprior, before learning the hyperparameter in hierarchical prior by using type-II maximum likelihood estimation. Thus the RVM_{BLSX} method implements the type-II maximum likelihood method to estimate the hyperparameter in the hierarchical prior wile Gibbs sampling to get the estimate hyperparameter in hyperprior.

Here is the structure for the following chapters: Chapter 2 of the thesis provides general background on sparse learning within the Bayesian framework. Chapter 3 will explore well-known sparse methods such as Lasso, Ridge, and Bayesian Lasso. In Chapter 4, we will delve into the development and procedure of two specific methods: RVM_{BLS} and RVM_{BLSX}. The former was developed by [Helgøy and Li, 2023], while the latter is my own contribution and serves as an extension of RVM_{BLS}. This chapter will offer a comprehensive walkthrough of both methods, going in-depth into the mathematics behind the models and the implementation algorithm. Chapter 5 is the result section, where we will compare each model on different benchmark datasets. The final conclusion and discussion for further work will be presented in chapter 6.

Chapter 2

General background

2.1 Sparse Modeling in supervised learning

As mentioned in chapter 1, when dealing with large datasets, algorithms that use all the available data can become slow and computationally expensive thus sparse learning methods is needed. The sparse learning in this thesis is within the framework of supervised learning. The sparse supervised learning methods can extract and identify the most informative samples and variables for further prediction with less computational burden, which can help us to interpret the model and patterns within the dataset better.

In supervised learning, we are given a dataset with n observations with input vectors $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})^T \in \mathbb{R}^p$ is the i'th observation's input vector (generally denoted as $\mathbf{x}$) containing observed values from p input variables. The n observed values for the scalar target is denoted as $\{t_i\}_{i=1}^n$ [Tipping, 2001]. From this dataset, we aim to construct and learn a model to study dependency of the target variable on the inputs with the objective of making accurate prediction of target variable y, especially for the previously unseen values of $\mathbf{x}$. Often we can build the model upon some function $y(\mathbf{x})$ in the following form of [Tipping, 2001] to approximate the relationship of input variables and target variable:

$$\begin{gathered} y(\mathbf{x};\mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}), \\ t = y + \varepsilon, \quad \varepsilon \sim \text{i.i.d.} \mathcal{N}(0, \sigma^2) \end{gathered} \tag{2.1}$$

where the output can be approximated as a weighted sum of M, generally nonlinear and fixed, basis functions of input variables as $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), .., \phi_M(\mathbf{x}))^T$. Analysis of equation (2.1) is facilitated since (2.1) is the linear combination of weighted basis functions where the weights are weight parameters $\mathbf{w} = (w_1, w_2, ..., w_M)^T$ appear linearly. One of the main objectives of learning is to estimate the values of those parameters based on certain optimization criteria.

In the field of machine learning, there is one well known supervised sparse learning method which utilize kernel basis functions, the support vector machine (SVM) ([Schölkopf et al., 2001], [Boser et al., 1992], [Vapnik et al., 1997]). The regression model in SVM makes prediction based on the function:

$$y(\mathbf{x};\mathbf{w}) = \sum_{i=1}^{n} w_i \mathbf{K}(\mathbf{x}, \mathbf{x}_i) \tag{2.2}$$

Where $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function utilized as a basis function. When defining one basis function for each individual sample's $\mathbf{x}_i, i = (1, .., n)$ in the training set, then the support vector machine based on equation (2.2) can be learned to be a sparse model which achieves sample size reduction, as only few of the estimated n sample weight parameters will be non-zero after the learning process. SVM model can avoid over-fitting as well as achieving good generalisation, and which furthermore results in prediction dependent only on a subset of non-zero weighted kernel function associated training samples $\mathbf{x}_l, (l = 1, .., L)$ with $(L < n)$ with L support vectors, and instead of using the whole sample set that contains n samples. Thus the SVM extract only the most informative sample for predictions and can be viewed as sample size reduction method. This thesis will introduce RVM$_{BLS}$ [Helgøy and Li, 2023] in the framework of sample size reduction method, while however we will concentrate most on the sparse method which deal with high dimensional data, so that the most important input variables can be selected for prediction. My own extension RVM$_{BLSX}$ is illustrated in the framework of dimensional reduction or variable selection learning.

From the point of a statistical view, the term high dimensional data is often used to refer to dataset, where the number of input variables (p) is near or exceeds the number of observations (n). When it comes to model such data, there are several potential limitations that must be considered. One such limitation is the curse of dimensionality [Bellman, 1966], which arises when the number of variables increases, then the number of observations required to avoid significant bias increases exponentially. Thus, it is often the case that there are not enough observations in high-dimensional data to account for all the variables present. Additionally, when modeling with high dimensional data, overfitting can occur, leading to models that are too complex and capture random noise in the data into the model structure[Bellman, 1966]. To mitigate these issues, models that perform variable selection or dimensionality reduction are necessary. These models aim to identify the most important features that impact the predicted output variable, resulting in more parsimonious models with better generalization capabilities. By utilizing sparse methods which can achieve variable selection, we can create more efficient and effective models that account for high dimensional data.

To summarize, variable selection is one type of sparse statistical method used to identify and select a subset of important variables from a large set of candidate variables of the high dimensional data for more precise prediction of output variables in a statistical model. The goal of variable selection is to improve the accuracy and interpretability of the statistical model by reducing the complexity of the model[Chowdhury and Turin, 2020], eliminating irrelevant or redundant variables as well as increasing the model's prediction precision.

2.2 Bayesian view of model learning

Even though the SVM is sparse and have good generalization property, from the view of Bayesian statistics, it has certain disadvantages in its methodology. Support Vector Machines (SVM) are known to produce a point estimate, which implies that the predictions made by this method are not probabilistic in nature. From a statistical point of view, it is desirable to obtain an estimate of the conditional distribution on given information on the data in order to quantify the uncertainty of the predictions. The posterior distribution which can combine the prior knowledge as well as given data information can achieve the goal to quantify the uncertainty of prediction. This is especially important in classification tasks where posterior probabilities of class membership play a crucial role in adapting to varying class priors and asymmetric misclassification costs. Another disadvantage is that the kernel function $K(\mathbf{x}, \mathbf{x}_i)$ used in SVM must satisfy Mercer's conditions, which state that it must be a continuous, symmetric kernel of a positive integral operator. While the more general kernel functions which do not satisfy Mercers conditions can be desired as basis function in certain cases[Tipping, 2001]. Tipping then developed a new sparse learning method in the Bayesian framework, which can be utilized as a more general type of kernel function and can also give output in probabilistic prediction, and this method is called the relevance vector machine (RVM) [Tipping, 2001]. RVM possesses a Bayesian view of model learning and we will give a short background introduction of sparse learning in the Bayesian view in the rest of this section.

When dealing with larger datasets, estimating parameters for **w** in equation (2.1) by using the frequentist statistical method such as ordinary least square regression (OLS) can be computationally intensive and time-consuming for each analysis. Moreover, for high dimensional data where the samples size of the dataset n is near the number of input variables p, estimation by use of the original OLS method can end up in estimated parameters that possess high variance. As will be illustrated later, OLS fails in high dimensional multivariate regression where p exceeds n. Then penalized regression learning is needed by including additional penalty on the coefficients to least squares loss function. The Bayesian approach however, places a sparse prior distribution directly on the weight parameter **w**. While frequentist modeling deals with uncertainty in data by using only the additional random term such as ε in (2.1) to account for noise and random errors, Bayesian modeling additionally seeks to capture uncertainty in models and associated parameters. This is accomplished by incorporating prior knowledge and treating parameters **w** as random variables. Through this approach, we can gain greater insight to quantify the uncertainty surrounding model parameters and final prediction.

In frequentist statistics, we assume the existence of a vector of unknown but fixed parameters **w** and aim to estimate them as accurately as possible using certain criteria. In contrast, a Bayesian approach does not assume the existence of a single true value for **w**, but rather seeks to identify a distribution of the parameters which is called the posterior distribution, calculated from likelihood of the observed data and prior distribution of the parameter. The calculation of the posterior distribution will utilize the following Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.3}$$

where A and B denotes random events. By using Bayesian theorem we obtain the posterior distribution over the parameters as:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}} \tag{2.4}$$

In equation (2.4), to determine the likelihood of observing the current data denoted as **t**, we define the probability $p(\mathbf{t}|\mathbf{w})$, where **w** represents the model parameters. Furthermore, we specify a prior distribution for the parameters, which reflects our prior beliefs or expectations about the parameter before any observations are made. This distribution is denoted as $p(\mathbf{w})$.

By denoting $p(\mathbf{w}|\mathbf{t})$ as the posterior distribution for **w**, the corresponding concrete statistical formula of (2.4) is:

$$p(\mathbf{w}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{t})} \tag{2.5}$$

After getting the posterior distribution, we can predict new data points $\mathbf{t}^*$ in a probabilistic way by using the following predictive distribution:

$$p(\mathbf{t}_*|\mathbf{t}) = \int p(\mathbf{t}_*|\mathbf{w})p(\mathbf{w}|\mathbf{t})d\mathbf{w} \tag{2.6}$$

As we marginalize over the parameter **w**, the predictive distribution relies solely on the observed data **t**, without the need for any additional information within the Bayesian framework. Moreover, Bayesian learning can provide a measure of uncertainty in the predictions. The key advantage of Bayesian learning, particularly in our scenario, is the ability to obtain a complete predictive and posterior distribution, instead of only providing a deterministic point estimate as with a fully deterministic approach. Further more, when a sparse prior is set for the **w**, the posterior combines the dataset information represented by the likelihood together with the prior information of parameter and end up in a posterior distribution such as the weight parameter **w** in (2.1) will spike at zero, and thus sparsity is achieved.

Chapter 3

Common Sparse Learning Methods

3.1 Multiple Linear Regression

As mentioned in chapter 1 and 2, the sparse learning methods introduced in this thesis are in the framework of supervised learning. This chapter begin with the most simple supervised learning model, the multiple linear regression, to introduce the mathematical framework needed for further sparse models [Douglas C. Montgomery, 2013]. Multiple linear regression is a model that can be used to predict the output variable Y based on the values of a set of independent input variables denoted as $\mathbf{X} = (X_1, ..., X_p)$. The multiple linear regression model has the ability to measure the relative contribution of each independent variable in the explanation or prediction of the output variable. The most simple multiple linear regression assumes a linear relationship between Y and $\mathbf{X}$. When assuming the intercept term is zero, the multiple linear regression formula can be denoted as follows:

$$Y = \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{3.1}$$

Thus equation (3.1) is the linear combination of the weighted variables while equation (2.2) can be viewed as linear combination of the weighted kernel functions where the kernel function is associated with samples. The slope coefficients for the predictor variables in (3.1) are denoted by β_k. Specifically, β_k represents the true average increase in the response variable Y associated with a one-unit increase in the value of predictor variable X_k, where k ranges from 1 to p, while holding the values of all other predictor variables constant. Thus, the slope coefficients provide a measure of the marginal effects of each predictor variable on the response variable, while controlling for the other predictor variables in the model.

Multiple linear regression enables us to make direct predictions based on information from p input variables simultaneously, while also allowing us to observe the unique effect of each predictor on Y [Douglas C. Montgomery, 2013]. The unknown parameter vector is denoted by $\beta = (\beta_1, ..., \beta_p)^T$, and the corresponding estimation is denoted by $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)^T$. β_j represents the true average increase in Y when X_k (where $k = 1, ..., p$) is increased by one unit while the values of all other variables are constant. After obtaining the estimation $\hat{\beta}$, the fitted value for Y is $\hat{Y} = \hat{\beta}_1 X_1 + ... + \hat{\beta}_p X_p$, where $e = Y - \hat{Y}$ is called the residual. Residuals are estimates of the true random error ε. The most common frequentist statistical way to estimate the parameter β is to obtain the estimation that minimizes the sum of squared residuals, also known as the Residual Sum of Squares, simplified as RSS. We can rewrite Y as a vector and $\mathbf{X}$ as a matrix after obtaining observations from n samples:

$$Y = (y_1, ... y_n)^T \tag{3.2}$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{3.3}$$

Now we can obtain the RSS(β) as a function of β and the corresponding first and second order derivatives of RSS(β) as follows:

$$\begin{aligned} \mathrm{RSS}(\beta) &= (Y - \mathbf{X}\beta)^T (Y - \mathbf{X}\beta) \\ \frac{d\mathrm{RSS}(\beta)}{d\beta} &= -2\mathbf{X}^T (Y - \mathbf{X}\beta) \\ \frac{d^2\mathrm{RSS}(\beta)}{d\beta d\beta^T} &= 2\mathbf{X}^T\mathbf{X} \end{aligned} \tag{3.4}$$

If we set the first derivative to equal zero, we obtain the estimation for the parameter vector β using the ordinary least squares (OLS) method, denoted as $\hat{\beta}^{OLS}$:

$$\hat{\beta}^{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y \tag{3.5}$$

Thus the RSS(β) is minimized when $\hat{\beta}^{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$ as the second order of derivative of $\mathrm{RSS}(\beta) = 2\mathbf{X}^T\mathbf{X}$ is larger than zero. $\hat{\beta}^{OLS}$ represents the estimated coefficients of β obtained through minimizing the residual sum of squares (RSS). The covariance matrix for $\hat{\beta}^{OLS}$ can be derived as follows:

$$\begin{aligned}\mathrm{Var}\hat{\beta}^{OLS} &= [(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathrm{Var}Y[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]^T \\ &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})((\mathbf{X}^T\mathbf{X})^{-1})^T \\ &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\end{aligned} \tag{3.6}$$

where:

$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{pmatrix} \zeta_1 & \cdot & \cdot & \cdot \\ \cdot & \zeta_2 & \cdot & \cdot \\ \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & \cdot & \zeta_p \end{pmatrix} \tag{3.7}$$

ζ refers to the diagonal elements of the matrix in (3.7). While minimizing $\mathrm{RSS}(\beta)$ with respect to β directly to obtain $\hat{\beta}^{OLS}$ in equation (3.5) may be appropriate for datasets with a sample size much larger than the number of variables, it may not suffice for more complex datasets in scenarios where the number of predictors p is approximately equal to the sample size n. When p is near n, it is common for some of the predictor variables to exhibit high correlation with each other. Consequently, the determinant of the product of the predictor matrix $\mathbf{X}$ and its transpose, $\mathbf{X}^T\mathbf{X}$, tends to approach zero, resulting in small values of ζ. This, in turn, leads to an increase in variance of $\hat{\beta}$ as:

$$\mathrm{Var}(\hat{\beta}_j) = \frac{\sigma^2}{\zeta_j} \tag{3.8}$$

Furthermore, in cases where the number of observations n is less than the number of predictors p, the predictor matrix $\mathbf{X}$ loses its ability to have linearly independent columns. This can result in a situation of 100% multicollinearity, where the columns of $\mathbf{X}$ become linearly dependent on each other. This, in turn, renders the product of $\mathbf{X}^T$ and $\mathbf{X}$ non-invertible, i.e., a singular matrix and OLS fails. To address the challenges posed by high-dimensional multiple regression problems, we introduce two widely used regularized models: LASSO[Tibshirani, 1996] and Ridge regression (Hoerl and Kennard [1970a] [Hoerl and Kennard, 1970b]).

3.2 Lasso

LASSO is a an acronym for "Least Absolute Shrinkage and Selection Operator[Tibshirani, 1996]. It is a popular regularized regression technique used in high-dimensional data analysis. It effectively performs variable selection by introducing a penalty term to $\mathrm{RSS}(\beta)$. In LASSO we estimate the β's by minimizing the following penalized RSS with respect to β [Tibshirani, 1996].

$$\begin{aligned}\mathrm{RSS}(\beta)_\lambda &= \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| \\ &= \mathrm{RSS}(\beta) + \lambda\sum_{j=1}^{p}|\beta_j|\end{aligned} \tag{3.9}$$

Minimizing the above RSS will result in certain estimated coefficients in equation (3.9) being exactly zero. We can rewrite (3.9) in the following formula:

$$\min_{\beta}\{(Y - \mathbf{X}\beta)^T(Y - \mathbf{X}\beta) + \lambda\sum_{j=1}^{p}|\beta_j|\}, \tag{3.10}$$

Assuming that the predictor matrix $\mathbf{X}$ is of full rank and can be standardized into an orthonormal matrix, such that $\mathbf{X}^T\mathbf{X} = I = (\mathbf{X}^T\mathbf{X})^{-1}$, the optimization problem of LASSO can be expressed as follows:

$$\begin{aligned}&\min_{\beta}\{Y^TY - Y^T\mathbf{X}\beta - \beta^T\mathbf{X}^TY + \beta^T\mathbf{X}^T\mathbf{X}\beta + \lambda\sum_{j=1}^{p}|\beta_j|\} \\ &\propto \min_{\beta} - \{[\hat{\beta}^{OLS}]^T\beta - \beta^T\hat{\beta}^{OLS} + \beta^T\beta + \lambda\sum_{j=1}^{p}|\beta_j\} \\ &= \min_{\beta_1,\dots,\beta_p}\{\sum_{j=1}^{p}(-2\hat{\beta}_j^{OLS}\beta_j + \beta_j^2 + \lambda|\beta_j|)\} \\ &= \sum_{j=1}^{p}(\min_{\beta_j} - 2\hat{\beta}_j^{OLS}\beta_j + \beta_j^2 + \lambda|\beta_j|)\end{aligned} \tag{3.11}$$

Let $\hat{\beta}^L$ denote the estimation of the lasso coefficient which can maximize (3.11). After certain calculation we can get if:

$$|\hat{\beta}_j^{OLS}| < \frac{\lambda}{2} \Leftrightarrow \begin{cases}(\hat{\beta}_j^{OLS} - \frac{\lambda}{2}) < 0 \\ (\hat{\beta}_j^{OLS}) + \frac{\lambda}{2} > 0\end{cases} \tag{3.12}$$

then $\hat{\beta}_j^L = 0$. Otherwise if $|\hat{\beta}_j^{OLS}| > \frac{\lambda}{2}$.

$$\hat{\beta}_J^L = \begin{cases} \hat{\beta}_j^{OLS} - \frac{\lambda}{2} & (\hat{\beta}_j^L > 0) \\ \hat{\beta}_j^{OLS} + \frac{\lambda}{2} & (\hat{\beta}_j^L < 0) \end{cases} \tag{3.13}$$

The result from equation (3.12) and (3.13) induces that the model achieves sparsity, which allows for variable selection in a linear model as the variables whose coefficients being 0 are deleted in the final estimated model. Consequently Lasso achieves as a sparse predictive model that is easy to interpret as it is only the most significant input variables are extracted to explain the relationship between input variables and output variable.

3.3 Ridge Regression

Ridge regression and LASSO are both regularization techniques used in linear regression to prevent overfitting. However, they differ in the type of penalty term added to RSS(β). Instead of using L_1 penalty term in the way of being the sum of absolute value for the coefficients, ridge regression uses an L_2 penalty term, which is the sum of the squares of the coefficients ([Hoerl and Kennard, 1970a],[Hoerl and Kennard, 1970b],[Melkumova and Shatskikh, 2017]).The effect of this penalty is to shrink the magnitude of the coefficients towards zero, but they are never exactly zero. This can help reducing the impact of multicollinearity, where predictors are highly correlated with each other. More concretely, ridge gets the estimation of β by minimizing the following penalized RSS with respect to β:

$$\min_{\beta} \{ (Y - \mathbf{X}\beta)^T (Y - \mathbf{X}\beta) + \lambda \sum_{j=1}^{p} \beta_j^2 \} \tag{3.14}$$

The strength of the penalty term is controlled by the tuning parameter λ. When λ is set to zero, the Ridge regression estimate reduces to the standard linear regression estimate. On the other hand, when λ is set to infinity, all the coefficients are shrunk towards zero, resulting in a constant term estimate of zero ($\hat{\beta}^{\text{ridge}} = 0$). For intermediate values of λ, the Ridge regression method balances the trade-off between shrinking the coefficients and fitting a linear model. This results in a set of coefficients that are smaller in magnitude compared to the linear regression coefficients, while still maintaining a good fit to the data. The fundamental objective behind incorporating the penalty term in Ridge Regression is to mitigate overfitting by shrinking the estimated coefficients([Hoerl and Kennard, 1970a],[Hoerl and Kennard, 1970b],[Melkumova and Shatskikh, 2017]), thereby reducing the variance of the model.

Owing to the effect of shrinkage, Ridge Regression is unable to assign coefficients of exactly zero to any variables. Therefore, it does not perform variable selection. Instead, Ridge Regression optimizes the coefficients to achieve a balance between fitting the data well and reducing the impact of multicollinearity among the predictor variables. Due to the inclusion of the penalty term in the objective function, ridge regression reduces the impact of the predictor variables without excluding them from the model. This results in a model that fits the data better and handles multicollinearity, but it doesn't perform variable selection.

3.4 Interpretation of LASSO in a Bayesian point of view

If we want to look at LASSO from a Bayesian standpoint, we would need to assume the usual linear model with normal distributed random errors and combine it with a specific prior distribution which can impose sparsity for the coefficient parameters β. For LASSO this distribution is a double exponential (Laplace) distribution with mean zero and a scale parameter.

Start with a simple model represented directly by observations,

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i, \quad i = (1, ..., n) \tag{3.15}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, and the residual $y_i - \hat{y}_i = \delta_i$. Then the likelihood data for the data would be[James et al., 2013a]:

$$\mathcal{L}(Y|X,\beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\delta_i^2}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\delta_i^2\right) \tag{3.16}$$

Assuming that we know that the following Laplace prior for parameter vector β is:

$$p(\beta) = (1/2b)\exp(-|\beta|/b) \tag{3.17}$$

The posterior for β would be calculated:

$$p(\beta|X,Y) \propto \mathcal{L}(Y|X,\beta)p(\beta|X) = \mathcal{L}(Y|X,\beta)p(\beta) \tag{3.18}$$

And then by substitution of the prior and the likelihood functions we get:

$$\begin{aligned}p(\beta|X,Y) &= \mathcal{L}(Y|X,\beta)p(\beta)\\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2\right)\left[\frac{1}{2b}\exp\left(-\frac{|\beta|}{b}\right)\right]\end{aligned} \tag{3.19}$$

The LASSO estimate is actually the mode for β of the posterior distribution in (3.19) when the LASSO solution is fulfilled. To prove this, we can rearrange the expression and show:

$$\begin{aligned}\mathcal{L}(Y|X,\beta)p(\beta) &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2\right)\left[\frac{1}{2b}\exp\left(-\frac{|\beta|}{b}\right)\right]\\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \left(\frac{1}{2b}\right)\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2 - \frac{|\beta|}{b}\right)\end{aligned} \tag{3.20}$$

Now we take the natural logarithm of the product so that we can simplify the equation.

$$\ln\left[\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \left(\frac{1}{2b}\right)\right] - \left(\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2 + \frac{|\beta|}{b}\right) \tag{3.21}$$

This makes it easier to formulate a strategy, which is to maximize the whole expression:

$$\underset{\beta}{\text{maximize}}\left\{\ln\left[\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \left(\frac{1}{2b}\right)\right] - \left(\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2 + \frac{|\beta|}{b}\right)\right\} \tag{3.22}$$

In order to maximize the whole expression, we need to minimize the second term of it:

$$\begin{aligned}&\arg\min_{\beta}\left(\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2 + \frac{|\beta|}{b}\right) = \arg\min_{\beta}\left(\frac{1}{2\sigma^2}\sum_{i=1}^n \delta_i^2 + \frac{1}{b}\sum_{j=1}^p |\beta_j|\right)\\ &= \arg\min_{\beta}\frac{1}{2\sigma^2}\left(\sum_{i=1}^n \delta_i^2 + \frac{2\sigma^2}{b}\sum_{j=1}^p |\beta_j|\right) = \arg\min_{\beta}\left(\sum_{i=1}^n \delta_i^2 + \lambda\sum_{j=1}^p |\beta_j|\right)\\ &= \arg\min_{\beta}\left(\text{RSS}(\beta_\lambda) + \lambda\sum_{j=1}^p |\beta_j|\right)\end{aligned} \tag{3.23}$$

By substituting $\lambda = \frac{2\sigma^2}{b}$ in place of b in equation (3.23), the process of solving the optimization problem presented in equation (3.23) is equivalent to minimizing equation (3.10). In other words, the optimization problem in equation (3.10) can be viewed as the search for the mode of the posterior in equation (3.23).

3.5 Bayesian Lasso

Section 3.4 demonstrates that the estimation of the Lasso parameter can be interpreted as a Bayesian posterior mode estimate, assuming independent double-exponential distributions as priors on the regression parameters. However, [Park and Casella, 2008] highlight that employing the prior from (3.17) results in the presence of multiple posterior modes for joint posterior distribution of β and σ^2, which introduces conceptual and computational challenges. They further introduce the Bayesian Lasso model, where a conditional Laplace prior is utilized for β. Here, we rephrase (3.15) in the following manner:

$$Y = \mathbf{X}\beta + \varepsilon_{,} \tag{3.24}$$

In this model, $\beta = (\beta_1, ..., \beta_p)^T$ while Y is the $n \times 1$ vector of the observed response values. $\mathbf{X}$ is the standardized regressors with $n \times p$ matrix of observed values for input variables, and ε is the $n \times 1$ vector of i.i.d. normal errors with mean 0 and unknown variance σ^2[Park and Casella, 2008].

As mentioned in section 3.3 the Lasso estimates are viewed as L_1-penalized least square estimates, and the minimization problem in Lasso is:

$$\min_{\beta}\{(\mathbf{Y}-\mathbf{X}\beta)^T(\mathbf{Y}-\mathbf{X}\beta)+\lambda\sum_{j=1}^{p}|\beta_j|\} \tag{3.25}$$

As also mentioned in section 3.3, the Lasso estimates can be interpreted as the posterior mode estimates when the regression parameters have i.i.d. Laplace priors which is unconditional on σ^2.

$$\pi(\beta) = \prod_{j=1}^{p}\frac{\lambda}{2}e^{-\lambda|\beta_j|} \tag{3.26}$$

By utilizing the unconditional prior from (3.26), the joint posterior distribution of β and σ^2 have more than one mode. The presence of multiple modes in the posterior distribution can significantly slow down the convergence of the Gibbs Sampler and diminish the interpretability of point estimates. [Park and Casella, 2008] considers a full Bayesian analysis using a conditional Laplace prior which is conditional on the variance σ^2:

$$\pi(\beta|\sigma^2) = \prod_{j}^{p}\frac{\lambda}{(2\sqrt{\sigma^2})}e^{-\lambda|\beta_j|\sqrt{\sigma^2}} \tag{3.27}$$

With the non-informative scale-invariant marginal prior $\pi(\sigma^2) = \frac{1}{\sigma^2}$. Conditioning on σ^2 is crucial in (3.27) to achieve a unimodal full posterior instead of getting multi-mode by using the unconditional prior in (3.26). Furthermore the Gibbs sampler implementation is utilized to sample the posterior distribution of β, σ^2 and to sample hyperparameter λ in (3.27). [Park and Casella, 2008].

Bayesian Lasso, unlike traditional Lasso regression, does not inherently guarantee sparsity in the estimated coefficients [Park and Casella, 2008]. Instead, it employs a prior distribution on the regression coefficients that encourages sparsity, resulting in some coefficients being shrunk towards zero. However, not all coefficients will be exactly zero. The degree of shrinkage is determined by the hyperparameter λ, which control the strength of the prior distribution. In contrast, traditional Lasso regression imposes a penalty on the sum of absolute values of the regression coefficients, directly promoting sparsity by setting many coefficients to exactly zero. Therefore, while Bayesian Lasso can promote sparsity in the estimates, it does not guarantee a fully sparse solution.

3.6 Sparse Bayesian Learning to reduce sample size n

This section will give a short presentation of probabilistic Sparse Bayesian Learning to reduce sample size n and extract the most important samples for prediction. As stated in Chapter 2, while SVM kernel-based sparse learning methods provides deterministic predictions, the probabilistic sparse learning method called relevance vector machine (RVM) introduced by [Tipping, 2001], offers a different approach which is in the framework of Bayesian inference and can also achieve sparse learning and probabilistic prediction. Our focus is on presenting RVM [Tipping, 2001] as a kernel-based method for reducing the sample size and do sample selection based on equation (2.2):

$$y(\mathbf{x};\mathbf{w}) = \sum_{i=1}^{n} w_i \mathbf{K}(\mathbf{x},\mathbf{x}_i) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,\sigma^2) \tag{3.28}$$

This is the same as equation (2.2) but with an added noise term ε. Thus RVM define $M = n$ in equation (2.1) and $\phi_j(\mathbf{x}) = K(\mathbf{x},\mathbf{x}_j), (j = 1,2,...,n)$ where $K(\cdot,\cdot)$ is a kernel function. Each weight parameter in $\mathbf{w}$ and basis function are associated with one individual sample's input vector in a training dataset. After the learning process, most of the estimated value of the weight parameters $\mathbf{w} = (w_1, .., w_n)^T$ will be close to zero or strictly zero. What remaining is the relevance input vectors corresponding to the non-zero weights, and those relevance vectors instead of the whole dataset, are left for model construction and prediction. Further more, RVM places an parameterized "automatic relevance determination" (ARD) Gaussian prior on the weight parameters $\mathbf{w}$ [Tipping, 2001]:

$$p(\mathbf{w}|\alpha) = \prod_{i=1}^{n} \mathcal{N}(w_i|\alpha_i^{-1}) \tag{3.29}$$

where $\alpha = (\alpha_1, ..., \alpha_n)^T$ denoting the inverse of the variances for each weight which can be viewed as precision hyperparameters, so that we obtain a individual hyperparameter α_i associated independently with weight w_i. RVM will estimate those hyperparameters by using type-II maximum likelihood method. RVM use a gamma distribution as a hyperprior for α and for the inverse of the variance for random error $\delta = \frac{1}{\sigma^2}$:

$$\begin{aligned} p(\alpha) &= \prod_{i=1}^{n} \frac{b^a}{\Gamma(a)} \alpha_i^{a-1} e^{-b\alpha_i} \\ p(\delta) &= \frac{d^c}{\Gamma(c)} \delta_i^{c-1} e^{-d\delta} \end{aligned} \tag{3.30}$$

The parameters above can often be set to zero in order to obtain uniform hyperpriors such that $a = b = 0$. If this is the case will be the basic prior for the weights become the improper prior[Tipping, 2001]:

$$p(w_i) \propto \frac{1}{|w_i|} \tag{3.31}$$

From [Tipping, 2001] we know that (3.31) can enforce sparsity also as it sharply peaked at 0. To maximize the marginal likelihood with respect to the hyperparameter α, the learning process for the RVM employs the type-II maximum likelihood method. During estimation, many values of α are set to infinity, resulting in corresponding posteriors for $\mathbf{w}$ being sharply peaked at zero and the corresponding estimated $\mathbf{w}$ values are set as 0. As a result, only a small number of relevance vectors that correspond to non-zero weights remain in the dataset, achieving sparsity by using the most informative samples with "relevant input vectors" for prediction [Tipping, 2001].

3.6.1 Type-II Maximum Likelihood Estimation

This section gives a short description of type-II maximum likelihood estimation. Type-II maximum likelihood, also known as empirical Bayes or evidence approximation[Jamil and Ter Braak, 2012], is a method for estimating hyperparameters in Bayesian hierarchical models. In Bayesian inference, the hyperparameters can represent the parameters in the prior distribution, which can be used to model uncertainty about the parameters[Tipping and Faul, 2003]. Type-II maximum likelihood, involves estimating the hyperparameters by maximizing the marginal likelihood of the observed data, averaged over all possible values of the parameters[Jamil and Ter Braak, 2012]. This method allows the hyperparameters to be estimated from the data itself, rather than being fixed or chosen subjectively.

Given model where **t** represents the observed data, $p(\mathbf{t}|\theta)$ denotes the likelihood function for the model given θ, are assumed to be a unknown vector of all parameters. In Bayesian hierarchical models, we can identify a hierarchical prior distribution for the parameter θ conditional on the hyperparameters, denoted as α. We can also identify a hyperprior distribution for the hyperparameter. Mathematically, we can get a joint distribution of θ and α as:

$$p(\theta,\alpha) = p(\theta|\alpha)p(\alpha) \tag{3.32}$$

Here, $p(\theta|\alpha)$ denotes the hierarchical prior distribution of the parameters conditioned on the hyperparameters, and $p(\alpha)$ represents the prior distribution of the hyperparameters themselves. The marginal likelihood can be calculated by the following integrating process:

$$p(\mathbf{t}|\alpha) = \int p(\mathbf{t}|\theta)p(\theta|\alpha)d\theta \tag{3.33}$$

In equation (3.33), $p(\mathbf{t}|\theta)$ is the likelihood function associated with the model, describing the probability of the observed data given the parameter values for θ. By integrating out the parameter θ, weighted by the prior distribution $p(\theta|\alpha)$, we obtain the marginal likelihood of the data as function of hyperparameters.

To estimate the hyperparameters, we maximize the marginal likelihood with respect to α, seeking the values that yield the highest likelihood for the observed data [Jamil and Ter Braak, 2012]. This estimation is expressed as:

$$\hat{\alpha} = \arg\max_{\alpha} p(\mathbf{t}|\alpha) \tag{3.34}$$

By solving this optimization problem, we obtain the maximum likelihood estimates of the hyperparameters, denoted as $\hat{\alpha}$. The estimated $\hat{\alpha}$ is obtained by plugging it into the posterior distribution of θ for further probabilistic inference of θ. Numerical optimization methods are commonly employed to find the values of α that maximize the marginal likelihood. Type-II maximum likelihood helps to avoid overfitting and reduce bias in parameter estimates. However, it can be computationally expensive and requires careful selection of the prior distribution for the hyperparameters [Tipping and Faul, 2003].

Chapter 4

New Sparse Learning Methods

4.1 RVM$_{BLS}$

The RVM$_{BLS}$ method [Helgøy and Li, 2023] utilizes the hierarchical structure from the Bayesian Lasso and the type-II maximum likelihood method described in section 3.6.1 to achieve sparse Bayesian learning. This method can be applied to both sample size reduction and dimensionality reduction, but in this paper, we focus on the former in terms of algorithmic introduction and variable selection in the result. In the RVM$_{BLS}$ method, each individual weight parameter w_i is associated with an individual sample with an input vector $\mathbf{x}_i \in \mathbb{R}^p$. In the regular RVM, Tipping utilizes the ARD prior from (3.29) for $\mathbf{w}$. RVM$_{BLS}$ instead use another ARD prior introduced in the following section as a prior for $\mathbf{w}$. This prior is conditional on σ^2, similar to the approach used by [Park and Casella, 2008]. In this new model, each weight is associated with an independent individual hyperparameter, is associated with the variances of the weights, rather than the precision hyperparameter α from (3.29). Thus the estimated zero hyperparameters in the ARD prior of RVM$_{BLS}$ correspond to the associated weighted parameters peaked at zero, and lead to the vectors with zero weight being pruned. Since the prior for $\mathbf{w}$ is conditional on σ^2, we can analytically prove that these hyperparameters will be set to zero by a certain threshold, while the threshold will be estimated directly from the data and is controlled by σ^2. Usually, σ^2 reflects the level of noise in a dataset. Once the hyperparameters have been estimated, the mode of the posterior weight parameter $\mathbf{w}$ can be used as a point estimate for $\mathbf{w}$. When presented with a new input $\mathbf{x}^*$, a point prediction for the corresponding target output can be obtained as follows:

$$y^* = \sum_{i=1}^{n} \hat{w}_i \phi_i(\mathbf{x}^*) + \hat{w}_0 \tag{4.1}$$

Here, $\hat{\mathbf{w}} = (\hat{w}_1, \ldots, \hat{w}_n)^T$ represents the point estimation for sample weights. Since the hyperparameters represent the variances of the weights, only the estimated nonzero hyperparameters will result in nonzero $\hat{\mathbf{w}}$. As a result, the RVM$_{BLS}$ method achieves sample size reduction by using only those $\hat{\mathbf{w}}$ corresponded input sample vectors in prediction and model construction. In the following section, we will provide a more detailed explanation of the RVM$_{BLS}$ algorithm.

4.1.1 RVM$_{BLS}$ for sample size reduction

In RVM$_{BLS}$ the likelihood function of the data set $\mathbf{y}$ is given by [Helgøy and Li, 2023];

$$p(\mathbf{y}|\mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{y}|\Phi\mathbf{w}, \sigma^2), \tag{4.2}$$

Φ represents the kernel matrix which contain the matrix elements $\Phi_{1i} = 1$ for $n = (1, ..., n)$ and $\Phi_{mn} = K(\mathbf{x}_m, \mathbf{x}_n)$, $i = (2, ..., n+1); n = (1, .., n)$. the ARD prior of the weights RVM$_{BLS}$ follows the conditional prior from equation (4.2), and is written:

$$p(\mathbf{w}|\gamma, \sigma^2) = \prod_{i=0}^{n} \mathcal{N}(\mathbf{w}_i|\gamma_i, \sigma^2), \tag{4.3}$$

where $\gamma = (\gamma_0, ..., \gamma_n)^T$ is the individual hyperparameter associated with the variance for the weights. More precise, the full conditional prior on $\mathbf{w}$ conditional on σ^2:

$$p(\mathbf{w}|\gamma, \sigma^2) = \prod_{i=0}^{n} \mathcal{N}(\mathbf{w}_i|0, \gamma_i\sigma^2) \tag{4.4}$$

Here we see that the variance for $\mathbf{w}$ is the combination of both hyperparameter and the variance within the data. The prior used for γ is an exponential hyperprior:

$$p(\gamma|\lambda) = \prod_{i=0}^{n} \frac{\lambda}{2} exp(-\frac{\lambda\gamma_i}{2}) \tag{4.5}$$

The priors for σ^2 and λ is:

$$\begin{aligned} p(\lambda) &= \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} \\ p(\sigma^2) &= \frac{d^c}{\Gamma(c)} (\sigma^2)_i^{c-1} e^{-d\sigma^2} \end{aligned} \tag{4.6}$$

The prior for the model parameters follows a gamma and inverse gamma distribution, with $a = b = c = d = 0$. With the prior structure established, the posterior distribution of all the model parameters can be calculated given the observed data $\mathbf{y}$:

$$p(\mathbf{w}, \gamma, \sigma^2, \lambda | \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\gamma, \sigma^2) p(\gamma|\lambda) p(\lambda) p(\sigma^2)}{p(\mathbf{y})} \tag{4.7}$$

Where $p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\gamma, \sigma^2) p(\gamma|\lambda) p(\lambda) p(\sigma^2)$ is the joint distribution of parameters, hyperparameters and data. The predictive distribution can be utilized to obtain predictions for y^* when presented with a new test input x^*:

$$p(y * |\mathbf{y}) = \int p(y^*|\mathbf{w}, \sigma^2) p(\mathbf{w}, \gamma, \sigma^2, \lambda | \mathbf{y}) d\mathbf{w} d\gamma d\sigma^2 d\lambda \tag{4.8}$$

However to obtain $p(\mathbf{w}, \gamma, \sigma^2, \lambda | \mathbf{y})$ we need the following decomposition:

$$p(\mathbf{w}, \gamma, \sigma^2, \lambda | \mathbf{y}) = p(\mathbf{w}|\mathbf{y}, \gamma, \sigma^2) p(\gamma, \sigma^2, \lambda | \mathbf{y}) \tag{4.9}$$

The first term on the right side in (4.9) , which is the posterior of $\mathbf{w}$ conditional on $\mathbf{y}$ and other parameters $p(\mathbf{w}|\mathbf{y}, \gamma, \sigma^2)$, can be calculated analytically by using Bayes rule, and is a Gaussian distribution with the mean vector and covariance[Helgøy and Li, 2023]:

$$\begin{aligned} \mu &= \sigma^{-2} \Sigma \Phi^T y \\ \Sigma &= [\sigma^{-2} \Phi^T \Phi + \Lambda^{-1}]^{-1} \end{aligned} \tag{4.10}$$

The matrix Λ is defined as $diag(\gamma_i \sigma^2)$. To estimate γ, [Helgøy and Li, 2023] employ the similar type-II maximum likelihood estimation process from [Babacan et al., 2010] which involves maximizing the second term in the decomposition (4.9) $p(\gamma, \sigma^2, \lambda | \mathbf{y})$ with respect to each individual hyperparameter γ_i:

$$p(\gamma, \sigma^2, \lambda | \mathbf{y}) = \frac{p(\mathbf{y}, \gamma, \sigma^2, \lambda)}{p(\mathbf{y})} \propto p(\mathbf{y}, \gamma, \sigma^2, \lambda) \tag{4.11}$$

Thus, to obtain a type-II maximum likelihood estimate of γ, the joint distribution

$p(\mathbf{y},\gamma,\sigma^2,\lambda)$ can be maximized. This is done by integrating out the weight parameter $\mathbf{w}$, yielding the following expression:

$$\begin{aligned} p(\mathbf{y},\gamma,\sigma^2,\lambda) &= \int p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\gamma,\sigma^2)p(\gamma|\lambda)p(\lambda)p(\sigma^2)d\mathbf{w} \\ &= (\frac{1}{2\pi})^{n/2}|\mathbf{C}|^{-1/2}e^{-\frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}}p(\gamma|\lambda)p(\lambda)p(\sigma^2) \end{aligned} \tag{4.12}$$

where $\mathbf{C} = (\sigma^2\mathbf{I}_n + \Phi\Lambda\Phi^T)$. The log of $p(\mathbf{y},\gamma,\sigma^2,\lambda)$ is:

$$\begin{aligned} L = &-\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y} + n\log\frac{\lambda}{2} - \frac{\lambda}{2}\sum_i \gamma_i \\ &+ a\log b - \log\Gamma(a) + (a-1)\log\lambda - b\lambda \\ &+ c\log d - \log\Gamma(c) - (c+1)\log\sigma^2 - \frac{d}{\sigma^2} \end{aligned} \tag{4.13}$$

Due to the type-II maximum likelihood estimation by maximizing (4.13) with respect to γ, resulting in some γ_i values will be set to zero, and the corresponding basis function is pruned out from the model. Thus resulting in a sparse model. The process of estimating the hyperparameters can be achieved using the fast optimization algorithm described in the next section.

4.1.2 Fast optimization algorithm

The RVM is a popular choice for utilizing Bayesian framework for sparse supervised learning, but it does have some disadvantages that should be considered. One such disadvantage is the sensitivity of the RVM to hyperparameters, including the choice of kernel function and regularization parameter. Proper tuning of these hyperparameters is necessary to achieve good performance, but this can be challenging and time-consuming [Tipping, 2001]. Another potential issue with the RVM is its susceptibility to overfitting, especially when working with small training sets or noisy datasets. If overfitting occurs, the model's ability to generalize to new data may be poor [Tipping, 2001]. In addition, the maximization of the type-II likelihood can be slow when computing the initial iterations, which require computations on the order of $O(n^3)$. The RVM starts by including all basis functions in the model and then iteratively updates the hyperparameters while pruning some of the basis functions [Tipping, 2001]. [Tipping and Faul, 2003] fixes this by using the Fast optimization algorithm for sparse Bayesian Models. The normal case of this algorithm updates the parameter γ (which represent a vector of parameters) for all versions, whilst in Fast algorithm it only updates a single parameter γ_i for each iteration.

In order to obtain the derivatives of L with respect to each single hyperparameter γ_i we rewrite the formula for L as:

$$\begin{aligned} L(\gamma) = &-\frac{1}{2}[\log|\mathbf{C}_{-i}| + \mathbf{y}^T\mathbf{C}_{-i}^{-1} + \mathbf{y}\sum_{j\neq i}\gamma_i] \\ &+\frac{1}{2}[\log\frac{1}{1+\sigma^2\gamma_i s_i} + \frac{q_i^2\sigma^2\gamma_i}{1+\sigma^2\gamma_i s_i} - \lambda\gamma_i] \end{aligned} \tag{4.14}$$

where $s_i = \Phi_i^T\mathbf{C}_{-i}^{-1}$ and $q_i = \Phi_i^T\mathbf{C}_{-i}^{-1}\mathbf{y}$. The log likelihood function has now been decomposed to two parts, the first one, where γ_i and the corresponding ϕ_i is excluded, and the second part contains the terms that involve γ_i. The covariance matrix in the log likelihood for (4.14) can be decomposed as:

$$\begin{aligned} \mathbf{C} &= \sigma^2\mathbf{I} + \textstyle\sum_{m\neq i}\sigma^2\gamma_m\phi_m\phi_m^T + \sigma^2\gamma_i\phi_i\phi_i^T \\ &= \mathbf{C}_{-i} + \sigma^2\gamma_i\phi_i\phi_i^T \end{aligned} \tag{4.15}$$

$\mathbf{C}_{-i}$ denotes $\mathbf{C}$ without the basis function i. Applying the Woodbury identity on the covariance matrix will give us the inverse of the covariance matrix as:

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_i^{-1}\phi_i\phi_i^T\mathbf{C}_{-i}^{-1}}{\sigma^{-2}\gamma_i^{-1} + \phi_i^T\mathbf{C}_i^{-1}\phi_i} \tag{4.16}$$

and the determinant identity has been used to obtain the following decomposition of the determinant:

$$|\mathbf{C}| = |\mathbf{C}_{-i}||1+\sigma^2\gamma_i\phi_i^T\mathbf{C}_i^{-1}\phi_i| \tag{4.17}$$

Now onto the derivative of $L(\gamma)$ with respect to γ_i, where all other parameters are fixed:

$$\begin{aligned} \frac{dL(\gamma)}{d\gamma_i} &= \frac{1}{2}[-\frac{s_i}{\sigma^{-2}+\gamma_i s_i} + \frac{q_i^2\sigma^{-2}}{(\sigma^{-2}+\gamma_i s_i)^2} - \lambda] \\ &= -\frac{(\gamma_i k_1 + \gamma_i k_2 + k_3)}{2(\sigma^{-2}+\gamma_i s_i)^2} \end{aligned} \tag{4.18}$$

where $k_1 = \lambda s_i^2$, $k_2 = s_i^2 + 2s_i\lambda\sigma^{-2}$ and $k_3 = \sigma^{-2}(\lambda\sigma^{-2} + s_i - q_i^2)$. The numerator has a quadratic form while the denominator is always positive so that $\frac{dL(\gamma)}{d\gamma_i} = 0$ is satisfied at

$$\gamma_i = \frac{-(s_i^2 + 2s_i\lambda\sigma^{-2}) \pm \sqrt{\Theta}}{2\lambda_i^2} \tag{4.19}$$

where

$$\Theta = (s_i^2 + 2s_i\lambda\sigma^{-2})^2 - 4\lambda s_i^2\sigma^{-2}(\lambda\sigma^{-2} + s_i - q_i^2) \tag{4.20}$$

The solution for the equation for γ_i is similar to the corresponding expression from [Babacan et al., 2010], by analysing the terms we can see that if $q_i^2 - s_i < \lambda\sigma^{-2}$ then $\Theta < s_i^2 + 2s_i\lambda\sigma^{-2}$, and both solutions are for γ_i are negative. Since $\frac{L(\gamma)}{d\gamma_i}$ at 0 is negative, the maximum of $L(\gamma)$ occurs at $\gamma_i = 0$. When $q_i^2 - s_i > \lambda\sigma^{-2}$, there are two solutions. One negative and one positive since $\frac{L(\gamma)}{d\gamma_i}$ us positive when $\gamma_i = 0$ and negative at $\gamma_i = \infty$.
The positive solution for γ_i maximizes $L(\gamma)$. So the maximum of $L(\gamma)$ when all other components are fixed, is therefore obtained at:

$$\gamma_i = \begin{cases} \frac{-(s_i^2+2s_i\lambda\sigma^{-2})\pm\sqrt{\Theta}}{2\lambda_i^2} & \text{if } q_i^2 - s_i > \lambda\sigma^{-2} \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

If we want to estimate λ from this equation we can just use the equation from earlier

$$\begin{aligned} L = &-\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y} + n\log\frac{\lambda}{2} - \frac{\lambda}{2}\sum_i \gamma_i \\ &+ a\log b - \log\Gamma(a) + (a-1)\log\lambda - b\lambda \\ &+ c\log d - \log\Gamma(c) - (c+1)\log\sigma^2 - \frac{d}{\sigma^2} \end{aligned} \tag{4.22}$$

Take the derivative of (4.22) with respect to λ and set it to zero, we get:

$$\hat{\lambda} = \frac{2(n+a-1)}{\sum_i(\gamma_i + 2b)} \tag{4.23}$$

To estimate the value of σ^2, we can apply a similar approach. Upon closer examination, we observe that we can separate σ^2 from the other components in the equation involving $\mathbf{C}$. This separation can be expressed as $\mathbf{C} = \sigma^2\tilde{\mathbf{C}}$, where $\tilde{\mathbf{C}}$ represents $\mathbf{C}$ with the component σ^2 excluded. Based on this separation, we can obtain the estimate for σ^2 as follows:

$$\hat{\sigma}^2 = \frac{\mathbf{y}^T\tilde{\mathbf{C}}^{-1}\mathbf{y} + 2d}{n + 2 + 2c} \tag{4.24}$$

In the optimization algorithm, both s_i and q_i needs to be updated, where [Helgøy and Li, 2023] and [Babacan et al., 2010] show that:

$$\begin{aligned} s_i &= \phi_i^T \mathbf{C}^{-1} \phi_i = \frac{S_i}{1 - \gamma_i \sigma^2 S_i}, \\ q_i &= \phi_i^T \mathbf{C}^{-1} \mathbf{y} = \frac{Q_i}{1 - \gamma_i \sigma^2 S_i} \end{aligned} \tag{4.25}$$

where:

$$\begin{aligned} S_i &= \sigma^{-2} \phi_i^T \phi_i - \sigma^{-2} \phi_i^T \phi \Sigma \phi^T \phi_i \sigma^{-2} \\ Q_i &= \sigma^{-2} \phi_i^T \mathbf{y} - \sigma^{-2} \phi_i^T \phi \Sigma \phi^T \mathbf{y} \sigma^{-2} \end{aligned} \tag{4.26}$$

In the equations (4.26), Σ and ϕ represents those basis functions that are currently included in the model. As mentioned in [Babacan et al., 2010], both Σ and μ are updated effectively for each iteration process, as well as only one hyperparameter γ_i is updated. By considering only the basis functions currently included in the model, the computation becomes significantly faster compared to the initial scenario where all n basis functions exist. Furthermore from (4.21), [Helgøy and Li, 2023] pointed out that the criteria for setting $\gamma_i = 0$ is dependent on λ and the variance σ^2. [Helgøy and Li, 2023] also proved as σ^2 tends to infinity, the hyperparameters γ_i will be set to zero. The RVM$_{BLS}$ method takes advantage of this property by incorporating information about σ^2 to adaptability adjust the number of zero hyperparameters during the estimation process of γ. In contrast, methods like regular RVM do not consider σ in their calculations. This can result in a lack of distinction between actual signal information and noise, leading to only a small number of γ_i being set to zero. Consequently, the regular RVM method may struggle to effectively identify and eliminate irrelevant features from the model when dealing with very noisy dataset. The following algorithm outlines the suggested implementation of RVM$_{BLS}$[Helgøy and Li, 2023]:

Algorithm 1 The RVM with conditional Laplace priors

```
Initialize $\sigma^2$ to some value 0.01 for example
Initialize all $\gamma_i = 0$ and $\lambda = 0$
while convergence criteria are not met, do
    Choose a $\gamma_i$
    if $q_i^2 - s_i > \lambda\sigma^{-2}$ and $\gamma_i = 0$ then
        Add $\gamma_i$ to the model
    else if $q_i^2 - s_i > \lambda\sigma^{-2}$ and $\gamma_i > 0$ then
        Re-estimate $\gamma_i$
    else if $q_i^2 - s_i < \lambda\sigma^{-2}$ then
        Prune $i$ from the model (set $\gamma_i = 0$
    end if
    Update $\sigma^2$ using equation (4.24)
    Update $\lambda$ using equation (4.23)
    Update $\Sigma$ and $\mu$
    Update $s_i$ and $q_i$ using equation (4.25) and (4.26)
end while
```

4.1.3 Making Predictions based on the estimated sparse model

When the learning algorithm converge we are left with a $L(L < n+1)$ non-zero γ_i and each of them correspond to a relevance basis function and a relevance input vector from the training data. We denote the vector that contains those L non-zero γ_i as γ_{MP}, for any new input data $\mathbf{x}^*$ we can now make predictions based on the posterior of the weights while conditioning on γ_{MP} and σ^2. The predictive distribution (4.8) can be approximated by:

$$p(y^*|\mathbf{y},\gamma_{MP},\hat{\sigma}^2) = \int p(y^*|\mathbf{y},\gamma_{MP},\hat{\sigma^2})p(\mathbf{w}|\mathbf{y},\gamma_{MP},\hat{\sigma^2})d\mathbf{w} \tag{4.27}$$

The distribution is Gaussian and analytically tractable, the predictive mean and variance is given by:

$$\begin{aligned} y^* &= \phi(\mathbf{x}^*)\mu_{MP} \\ \sigma^{2*} &= \hat{\sigma}^2 + \phi(\mathbf{x}^*)^T\Sigma_{MP}\phi(\mathbf{x}^*), \end{aligned} \tag{4.28}$$

μ_{MP} and Σ_{MP} is calculated by:

$$\begin{aligned}\mu_{MP} &= \hat{\sigma}^2\Sigma_{MP}\Phi_{MP}^T\mathbf{y},\\ \Sigma_{MP} &= [\hat{\sigma}^2\Phi_{MP}^T + \Lambda_{MP}^{-1}]^{-1}\end{aligned} \tag{4.29}$$

In this setup $\phi(\mathbf{x}^*) = [\phi_1(\mathbf{x}^*), ..., \phi_{L.}(\mathbf{x}^*)]^T$ and $\phi_j(\mathbf{x}^*) = K(\mathbf{x}^*, \mathbf{x}_j), j = (1, ..., L)$ where $\mathbf{x}_j$ is the $j'th$ relevance input among the total L relevance input vectors within the total amount of relevance input vectors L. $\Phi_{MP} = [\phi_1, ..., \phi_L]$ is the $n \times L$ design matrix with the column vectors being $\phi_j = [\phi_1(\mathbf{x}_1), ..., \phi_1(\mathbf{x}_n)]^T$. The estimated diagonal matrix Λ_{MP} with elements $\hat{\sigma}^2\gamma_{MP}$ is a $L \times L$ matrix making μ_{MP} and Σ_{MP} the estimated posterior mean vector and covariance matrix over the weight. They only contain L non-zero elements that correspond to those non-zero elements in γ_{MP} [Helgøy and Li, 2023]. In a practical sense, the predictive mean is used for point prediction and the predictive variance can be used to construct a prediction interval.

4.1.4 Marginal Prior for RVM$_{BLS}$ and RVM

Here we give a simple comparison for the marginal prior in RVM$_{BLS}$ and RVM. Integrating out the hyperparameters γ in equation (4.5), we obtain the marginal prior for $\mathbf{w}$. This is given by:

$$p(\mathbf{w}|\sigma^2) = \int p(\mathbf{w}|\sigma^2, \gamma)p(\gamma)d\gamma = \prod_{i=0}^{n} \frac{\sqrt{\lambda}}{2\sqrt{\sigma^2}} e^{-\sqrt{\lambda}|w_i|/\sqrt{\sigma^2}} \tag{4.30}$$

The prior used is known as a Laplace prior conditioned on σ^2. Which is the same type of conditional prior used in bayesian Lasso as in equation (3.27) set on the weight parameter β for the variables, the conditional prior in (4.30) is set to the weight parameter $\mathbf{w}$. This prior can be derived from the Laplace distribution, which is a scaled mixture of Gaussians with an exponential mixing density [Andrews and Mallows, 1974]:

$$\frac{\sqrt{v}}{2} e^{-\sqrt{v}|x|} = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda}} e^{-x^2/(2\lambda)} \frac{v}{2} e^{-v\lambda/2} d\lambda \tag{4.31}$$

If we compare equation (4.31) with the marginal prior for $\mathbf{w}$ in RVM:

$$p(\mathbf{w}) = \int p(\mathbf{w}|\alpha)p(\alpha)d\alpha \tag{4.32}$$

the hyperparameter is the inverse of the variance for each weight, while marginalized the hyperprior will result in a student-t distribution.

$$\begin{aligned} p(\mathbf{w}_i) &= \int p(\mathbf{w}|\alpha_i)p(\alpha_i)d\alpha_i \\ &= \frac{b^a\Gamma(a+\frac{1}{2})}{(2\pi)^{\frac{1}{2}}\Gamma(a)}(b+\mathbf{w}_i^2/2)^{-(a+\frac{1}{2})} \end{aligned} \tag{4.33}$$

Both the Laplace Prior in (4.31) and student-t distribution in (4.33) is sparse[Tipping, 2001], which enhance the sparsity in both models.

4.2 RVM$_{BLSX}$

In this section, I will discuss my extension of the RVM$_{BLS}$, which I have tentatively named RVM$_{BLSX}$. The RVM$_{BLSX}$ method is developed to achieve variable selection ans is based on the multivariate linear regression model:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n), \tag{4.34}$$

RVM$_{BLSX}$ adopts the hierarchical structure in Bayesian Lasso proposed by [Park and Casella, 2008]:

$$\begin{aligned} \text{Likelihood } p(\mathbf{y}|\beta,\sigma^2) &= \mathcal{N}(\mathbf{y}|\beta\mathbf{X},\sigma^2) \\ \text{Hierarchical - Prior } p(\beta|\gamma,\sigma^2) &= \prod_{j=1}^{p}\mathcal{N}(\beta_j|0,\gamma_j\sigma^2) \\ \text{Hyper - Prior } p(\gamma|\lambda) &= \prod_{j=1}^{p}\frac{\lambda}{2}\text{exp}^{-\frac{\lambda\gamma_j}{2}} \end{aligned} \tag{4.35}$$

The Bayesian Lasso method employs a conditional marginal prior, which is obtained by integrating out the hyperparameter γ from the hierarchical prior from (4.35). This results in the following expression for the marginal prior:

$$p(\beta|\sigma^2) = \prod_{j=1}^{P}\frac{\lambda}{2\sigma^2}exp(-\lambda\frac{|\beta|}{\sigma^2}) \tag{4.36}$$

To sample the hyperparameter λ, the [Park and Casella, 2008] recommended utilizing a Markov Chain Monte Carlo Expectation-Maximization (MCEM) algorithm in combination with a Gibbs sampler. In the implementation of the MCEM, a λ value is estimated in each iteration of the algorithm, and this value is then used to run the next iteration of the Gibbs sampler.

Within the RVM$_{BLSX}$ we employ the same hierarchical structure that [Park and Casella, 2008] introduced for the regression model (4.34), where the weight parameter is the coefficient for the variable and $\mathbf{X}\beta$ is the linear combination of the variables $X_1,..,X_p$. In RVM$_{BLS}$ from section 4.1 the same type hierarchical structure is applied for the model (3.28) where the weight parameter is the sample weight, and $\phi(\mathbf{w})$ is the weighted combination of the kernel basis function $K(\mathbf{x},\mathbf{x}_i)$ where each kernel is associated with one sample. Since we employ the same hierarchical structure as the Bayesian Lasso, we should also estimate λ the same way as the Bayesian Lasso does. Both RVM$_{BLS}$ and RVM$_{BLSX}$ will utilize type-II maximum likelihood to estimate the hyperparameter γ in the ARD prior. In RVM$_{BLS}$ the hyperparameter λ in the hyperprior from (4.35) is estimated by maximizing the same marginal likelihood function directly to λ. Whereas in the original [Park and Casella, 2008] λ is sampled through Gibbs sampling. In this section we will calculate λ by using the same method that[Park and Casella, 2008] used in their paper. And the process is as follows:

(a) Let $k=0$ and choose initial $\lambda^{(0)}$

(b) Generate a sample from the posterior distribution of $\beta,\sigma^2,\gamma_1^2,...,\gamma_p^2$ using the Gibbs sampler with λ set to λ^k

(c) (E-step:) Approximate the expected "complete-data" log likelihood for λ by substituting averages based on the Gibbs sample of the previous step for any term involving $\beta,\sigma^2,\gamma_1^2,...,\gamma_p^2$

(d) (M-step:) Let $\lambda^{(k+1)}$ be the value of λ that maximises the expected log likelihood of the previous step

(e) Return to the second step and iterate until desired level.

After Gibbs sampling the complete log likelihood for λ by substituting averages based on Gibbs sampling it becomes.

$$-((n+p-1)/2+a+1)ln(\sigma^2)-\frac{1}{\sigma^2}((\tilde{\mathbf{y}}-\mathbf{X}\beta)^T(\tilde{\mathbf{y}}-\mathbf{X}\beta)/2+\gamma) \\ -\frac{1}{2}\sum_{j=1}^{p}ln(\gamma_j^2)-\frac{1}{2}\sum_{j=1}^{p}\frac{\beta_j^2}{\sigma^2\gamma_j^2}+p\,ln(\lambda^2)-\frac{\lambda}{2}\sum_{j=1}^{p}\gamma_j^2 \tag{4.37}$$

In this case, we omit certain additive constant terms that do not involve the parameter λ. To obtain the ideal E-step for the iterates at k, we calculate the expected value of the log likelihood conditioned on $\tilde{\mathbf{y}}$ under the current iterate λ^k.

$$Q(\lambda | + \lambda^k) = pln(\lambda^2)\frac{\lambda^2}{2}\sum_{j=1}^{p} E_{\lambda^{(k)}}[\gamma_j^2|\tilde{\mathbf{y}}] \tag{4.38}$$

Then the M-step becomes a simple analytical solution:

$$\lambda^{k+1} = \sqrt{\frac{2p}{\Sigma_{j=1}^{p} E_{\lambda(k)}[\gamma_j^2|\tilde{\mathbf{y}}]}} \tag{4.39}$$

The conditional expectations is replaced with the sample averages from the Gibbs samplers run.

The use of Monte Carlo techniques offers a practical means to approximate the likelihood function, making their implementation straightforward [Park and Casella, 2008]. By employing these techniques, we can estimate the likelihood function without the necessity of deriving explicit expressions for it [Park and Casella, 2008]. Let $\theta = (\beta, \sigma^2, \gamma_1^2, .., \gamma_p^2)$, allowing us to express the likelihood ratio for any λ as:

$$\begin{aligned}\frac{L(\lambda|\tilde{\mathbf{y}})}{L(\lambda_0|\tilde{\mathbf{y}})} &= \int \frac{L(\lambda|\tilde{\mathbf{y}})}{L(\lambda_0|\tilde{\mathbf{y}})}\pi_\lambda(\theta|\tilde{\mathbf{y}})d\theta = \int \frac{f_\lambda(\tilde{\mathbf{y}},\theta)\pi_{\lambda 0}(\theta|\tilde{\mathbf{y}})}{\pi_\lambda(\theta|\tilde{\mathbf{y}})f_{\lambda 0}(\tilde{\mathbf{y}},\theta)}d\theta \\ &= \int \frac{f_\lambda(\tilde{\mathbf{y}},\theta)}{f_{\lambda 0}(\tilde{\mathbf{y}},\theta)}d\theta\end{aligned} \tag{4.40}$$

The complete joint density above is denoted as f_λ for any particular given λ and π_λ is the full posterior of the model. Since the complete density is known for all λ we can use the final expression to approximate the likelihood ratio as a function of λ from a single Gibbs sample taken at the fixed λ_0:

$$\frac{f_\lambda(\tilde{\mathbf{y}},\theta)}{f_{\lambda 0}(\tilde{\mathbf{y}},\theta)} = (\frac{\lambda^2}{\lambda_0^2})^p \exp\left\{-(\lambda^2 - \lambda_0^2)\Sigma_{j=1}^{p}\frac{\gamma_j^2}{2}\right\} \tag{4.41}$$

Making it that the approximation in the neighbourhood of λ_0:

$$\frac{L(\lambda|\tilde{\mathbf{y}})}{L(\lambda_0|\tilde{\mathbf{y}})} = (\frac{\lambda^2}{\lambda_0^2})^p \int \exp\left\{-(\lambda^2 - \lambda_0^2)\Sigma_{j=1}^{p}\frac{\gamma_j^2}{2}\right\}\pi_{\lambda 0}(\gamma_1^2, ..., \gamma_p^2|\tilde{\mathbf{y}})d\gamma_1^2, .., \gamma_p^2 \tag{4.42}$$

The empirical Bayes approach of utilizing the marginal maximum likelihood estimate for λ

is an approach that does not automatically account for uncertainty in the maximum likelihood estimate [Park and Casella, 2008]. However, the effect of this uncertainty can be evaluated by considering the range of values that are contained within a 95 percent confidence interval where λ lies. The new algorithm for RVM$_{BLSX}$ consists of two algorithms which is as follows:

Algorithm 2 Gibbs sampler for Bayesian Lasso Empirical Bayes

```
Inputs: $x$ (Matrix of predictors), $y$ (Response vector), $n_{\max}$ (Number of iterations), $EB$ (Flag for empirical Bayes estimation), $a$ (Shape parameter), $b$ (Scale parameter), $print.it$ (Flag for printing iteration information)
Convert $x$ to a matrix and obtain the dimensions $n$ and $p$
Scale the data: subtract mean of $x$ and $y$ from each variable
Compute $X^T X$ and $X^T Y$
Initialize simulation matrices: beta.sim, sigma2.sim, tau2.sim, lambda.sim, llkhd
Initialize parameters: $\beta$, $\sigma^2$, $\gamma^2$, $\lambda$
if $EB$ is TRUE then
    Set $a = 0, b = 0$
    Define log-likelihood function $Q$
    Calculate initial log-likelihood $L_0$
else
    Set $\lambda$ (Lasso tuning parameter) to a desired value
end if
Set iteration counter $iter = 1$
while $iter < n_{\max}$ do
    Update $\beta$ by sampling from its full conditional distribution
    for $j = 1$ to $p$ do
        Update $\beta_j$ by sampling from its full conditional distribution and apply Lasso thresholding
    end for
    Update $\sigma^2$ by sampling from its full conditional distribution
    Update $\gamma^2$ by sampling from its full conditional distribution
    if $EB$ is TRUE then
        Update $\lambda$ based on empirical Bayes estimation
    end if
    if $iter >$ burn then
        Save current values of $\beta$, $\sigma^2$, $\gamma^2$, $\lambda$ for posterior statistics
    end if
    Increment $iter$ by 1
end while
Compute posterior statistics using saved samples
Return posterior statistics
```

Algorithm 3 The RVM with conditional Laplace priors and estimated λ

```
Initialize $\sigma^2$ to some value 0.01 for example
Initialize all $\gamma_i = 0$
Initialize all $\lambda$ through algorithm 2
while convergence criteria are not met, do
    Choose a $\gamma_i$
    if $q_i^2 - s_i > \lambda\sigma^{-2}$ and $\gamma_i = 0$ then
        Add $\gamma_i$ to the model
    else if $q_i^2 - s_i > \lambda\sigma^{-2}$ and $\gamma_i > 0$ then
        Re-estimate $\gamma_i$
    else if $q_i^2 - s_i < \lambda\sigma^{-2}$ then
        Prune $i$ from the model (set $\gamma_i = 0$)
    end if
    Update $\sigma^2$
    Update $\Sigma$ and $\mu$
    Update $s_i$ and $q_i$
    Update $\lambda$
end while
```

Algorithm 3 is nearly identical to algorithm 1. The initialization of λ is the key difference. The next chapter explores this further to see if the proposed model RVM$_{BLSX}$ can achieve sparsity and still contain strong predictive power.

Chapter 5

Results and Discussion

In this chapter we compare the models described in this thesis through several empirical analyses. We will compare the models by looking into the predictive performance as well as illustrate the sparsity of the models. We will look at examples from three benchmark datasets, the diabetes dataset from LARS[Efron et al., 2004], Boston housing and the Friedman #1 data.

Diabetes

The diabetes dataset in the "lars" package is a well-known dataset commonly used in regression analysis and machine learning. The dataset contains 442 diabetes patients measured on 10 variables which include age, sex, body mass index (BMI), average blood pressure, and six blood serum measurements. The response variable is a quantitative measure of disease progression one year after baseline[Efron et al., 2004].

Boston Housing

The Boston Housing Dataset is a widely used dataset in machine learning and statistics, often used as a benchmark for regression models. It was first introduced by Harrison and Rubinfeld in 1978 [Harrison and Rubinfeld, 1978] and is based on data collected from the Boston, Massachusetts area. The dataset consists of 506 samples, each representing a different suburb of Boston. For each suburb, 13 features are provided, including both numerical and categorical variables such as he per capita crime rate, average number of rooms per dwelling, proportion of residential land zoned for lots over 25,000 square feet, and others, and corresponding response variable medv which is the median value of occupied homes in 1000's.

Friedman #1 Data

The Friedman 1 dataset is a synthetic dataset or evaluating regression algorithms. It was introduced by Jerome H. Friedman in 1991 [Friedman, 1991]. The Friedman 1 dataset is generated based on a mathematical function that simulates a complex nonlinear relationship. It is designed to mimic real-world scenarios where the relationship between predictors and the target variable is nonlinear and exhibits interactions among the predictors. The dataset consists of 10 input features and a response variable. The input variables are random uniform input variables from [0,1]. The formula is:

$$y(\mathbf{x}) = 10\sin(\pi\mathbf{x}_1\mathbf{x}_2) + 20(\mathbf{x}_3 - 0.5) + 10\mathbf{x}_4 + 5\mathbf{x}_5 \tag{5.1}$$

The response variable is only dependent on the first five features while the remaining five features are noise columns. We use a sample of 200 when evaluating our models.

5.1 LASSO, Ridge and Bayesian Lasso

Benchmark datasets

In this section we will compare some of the results gathered from the models from section 3 in the thesis. In these examples, we divide the data into a training set and a test set, with 70% of the data allocated to the training set and the remaining 30% is the test set. We repeat this process 100 times with randomly chosen partitions of the training and the test set each time, and calculate the average RMSE of the test set across these repetitions. The result is printed in the following table:

Table 5.1: RMSE for methods used in Chapter 3

Dataset Method	Boston Housing	Diabetes	Friedman 1 Data
Lasso	4.563	51.278	2.25
Ridge	4.896	50.770	2.33
Bayesian Lasso	23.940	169.450	13.615

Both Lasso and Ridge produces similar results whilst the Bayesian Lasso produces obviously much larger root mean squared error (RMSE). In these three scenarios, the Bayesian Lasso have the worst performance.

To compare the estimated coefficient values of the variables in each model, I have chosen to illustrate the average value of each coefficient estimation in the diabetes dataset over 100 repetitions and the result is presented in figure 5.1, 5.3 and 5.6. We also present frequency plots in figures 5.2, 5.4, 5.5, 5.7 and 5.8 illustrates how many times each features is chosen over 100 repetitions.

LASSO

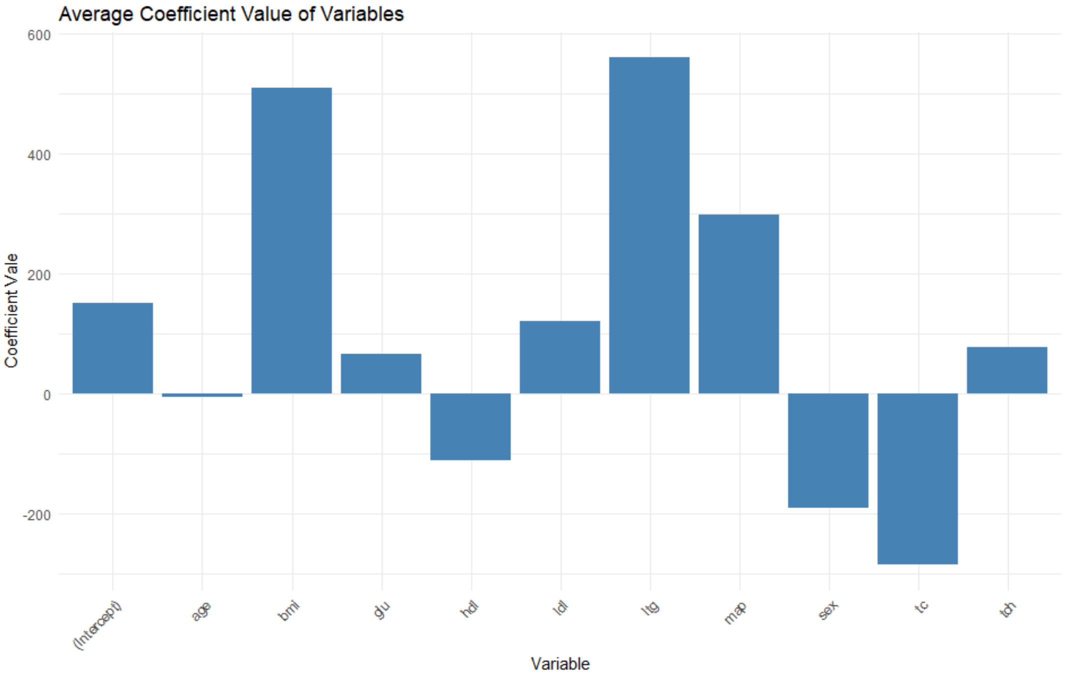

Figure 5.1: Average Coefficient Values for Lasso

From Figure 5.1 we can observe that there are are some coefficients as bmi and ltg that is much is larger than the coefficient variables for eg. age, which means the variable of bmi and ltg can be more influential than age when it come to the prediction of the output variable . If we look at the sparsity of the model, we can use the following figure which shows the frequency of each feature chosen in 100 repetitions:

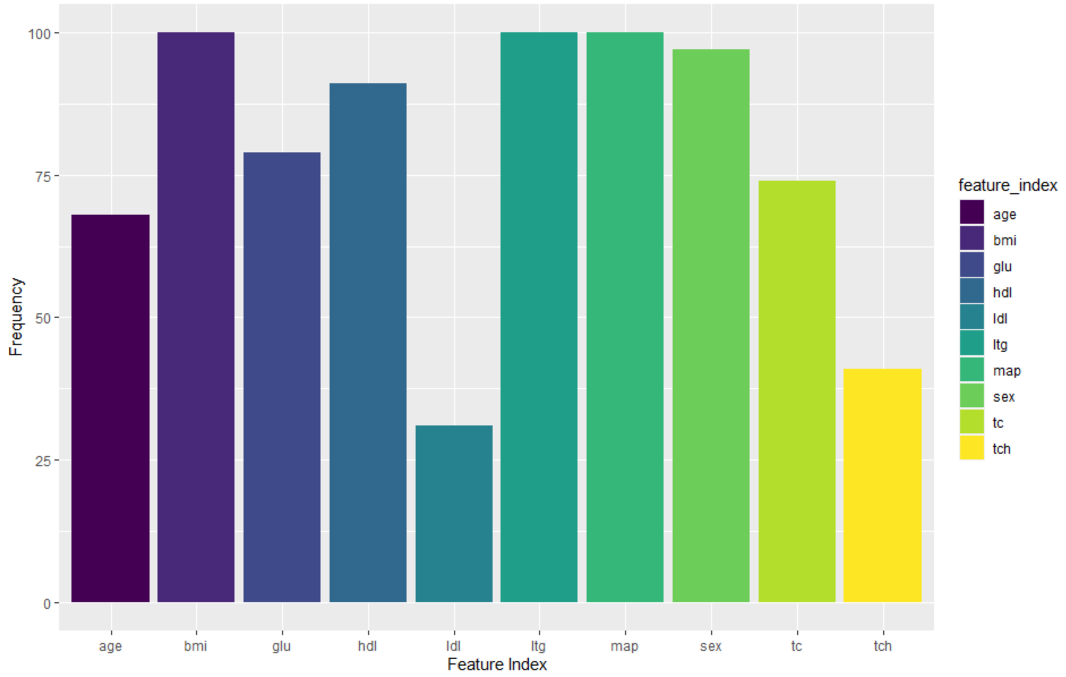

Figure 5.2: Chosen features over 100 repetitions Lasso

We can see from Figure 5.2 that the variables such as age and ldl gets chosen less then 70% of the times whilst features as map and bmi gets chosen almost 100% of the repetitions.

Ridge Regression

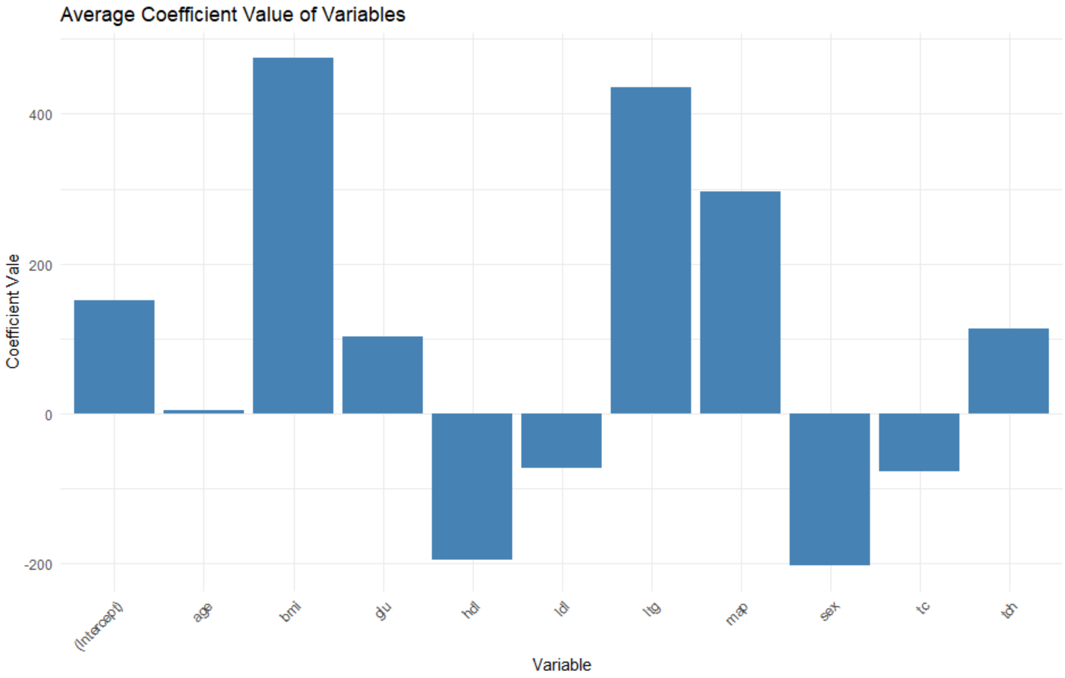

Figure 5.3: Average value of coefficients in 100 repetitions Ridge

If we should compare Figure 5.3 with Figure 5.1 we could say that they look very much alike. It is not identical, but the pattern is the same where the model values BMI the most and age the least. If we look at the sparsity of the model by using Figure 5.3:

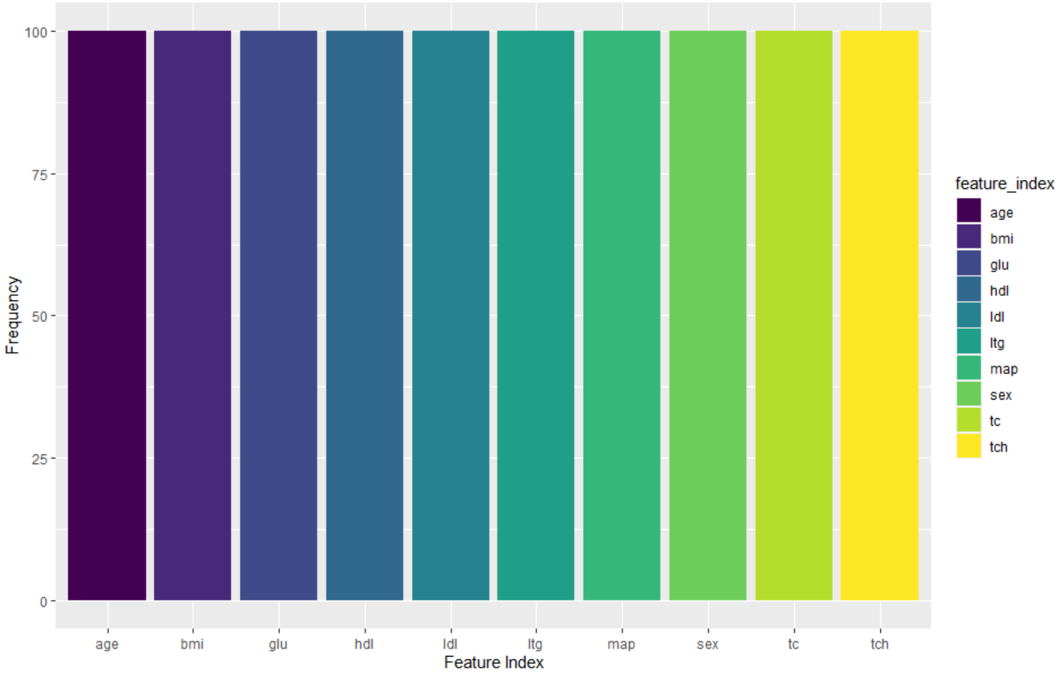

Figure 5.4: Chosen features over 100 repetitions Ridge

This model is not sparse at all, this is as we explained in section 3 that Ridge is not an automatically sparse model. If we adjust our model to have a balanced combination of L1 and L2 penalty term the result may differ. This is done within the package in R "glmnet" where we tune the α value to a number between 0-1. If we set the model to 1 it will be a Lasso model, if we set it to zero it will be as figure 5.4. Setting α to 0.5 produces this result:

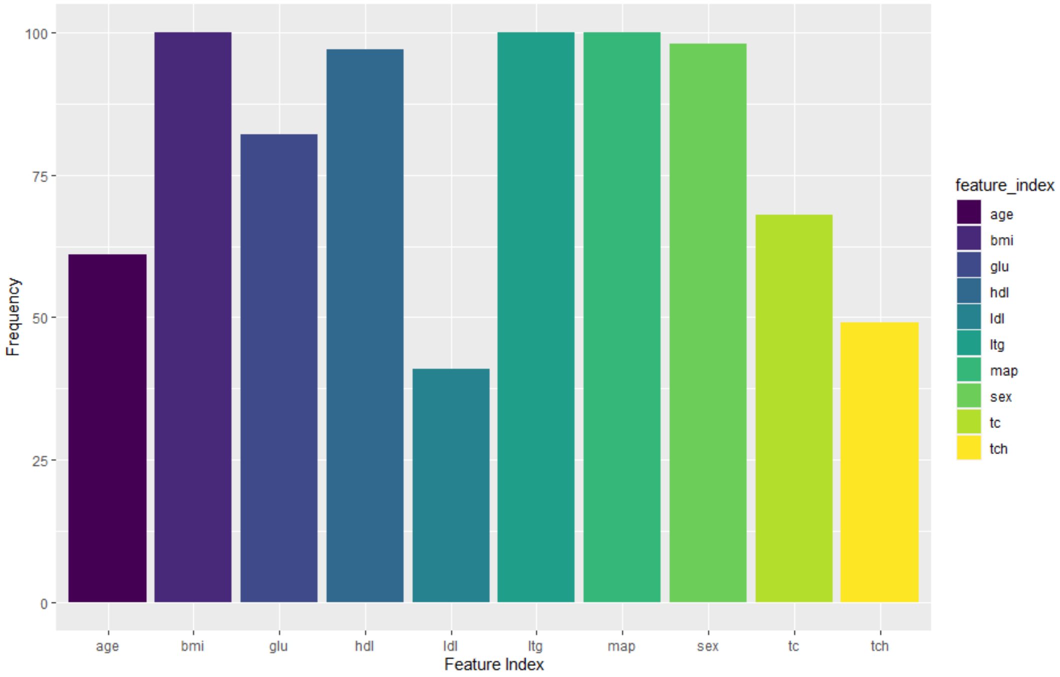

Figure 5.5: Chosen features over 100 repetitions Ridge Spars

Figure 5.5 gives us a more sparse model than 5.4, which was the desired outcome. This is called a elastic-net method which combines L1 and L2 regularization[Zou and Hastie, 2005] which can be described as a combination of the Ridge and Lasso model. This is not strictly a ridge regression model but is how we can achieve sparse results similar to the ridge regression.

Bayesian Lasso

For the Bayesian Lasso we will use the R package "monomvn". As mentioned in chapter 3, the original Bayesian Lasso can not achieve sparsity automatically, since none of the estimated coefficients is exactly zero.

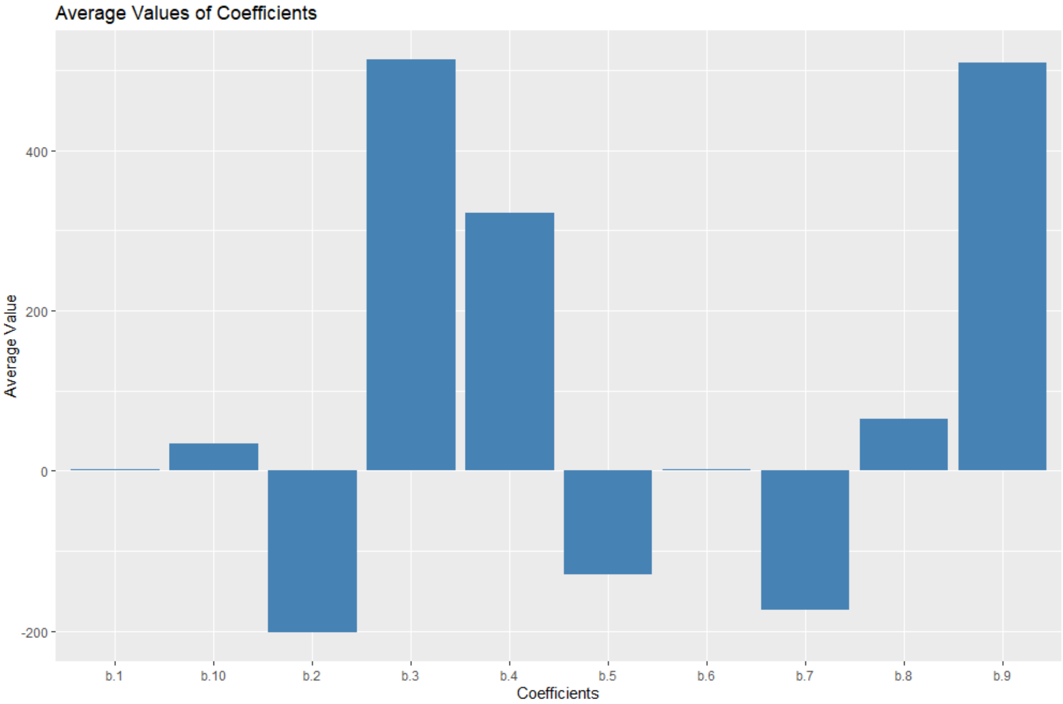

Figure 5.6: Chosen features over 100 repetitions Bayesian Lasso

Figure 5.6 shows us how the Bayesian Lasso estimation of the coefficients can be near zero, but never be exactly zero. Thus, the frequency table will be of the following kind:

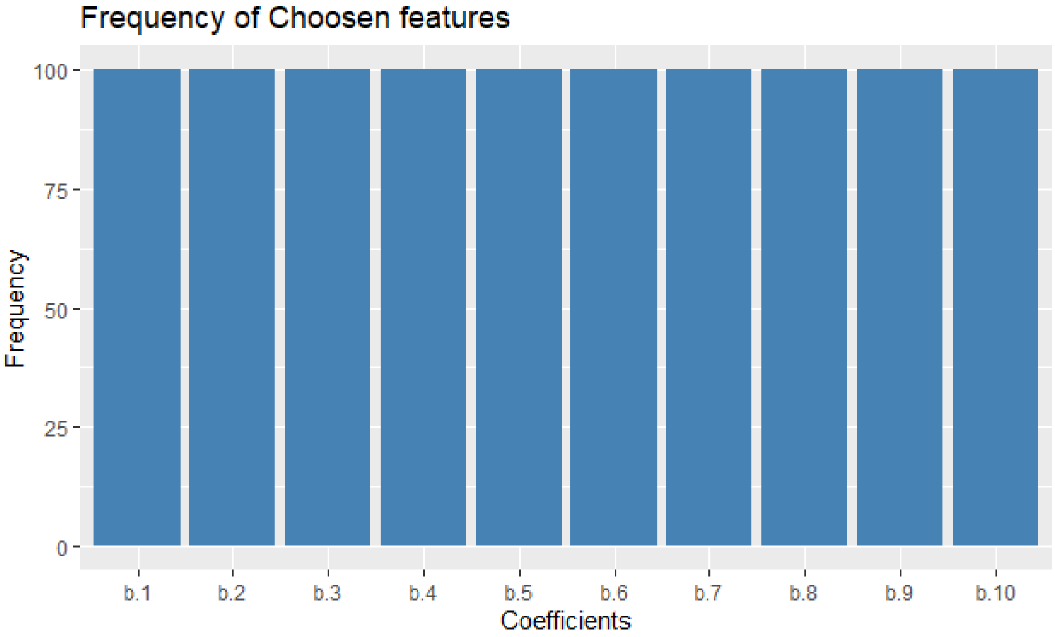

Figure 5.7: Chosen features over 100 repetitions Bayesian Lasso

Figure 5.7 shows that all features are included for each repetition. However, during the algorithm's implementation, a manual threshold is applied to the estimated coefficient values.

Any coefficient value below the threshold is set to zero. Resulting in a sparse model:

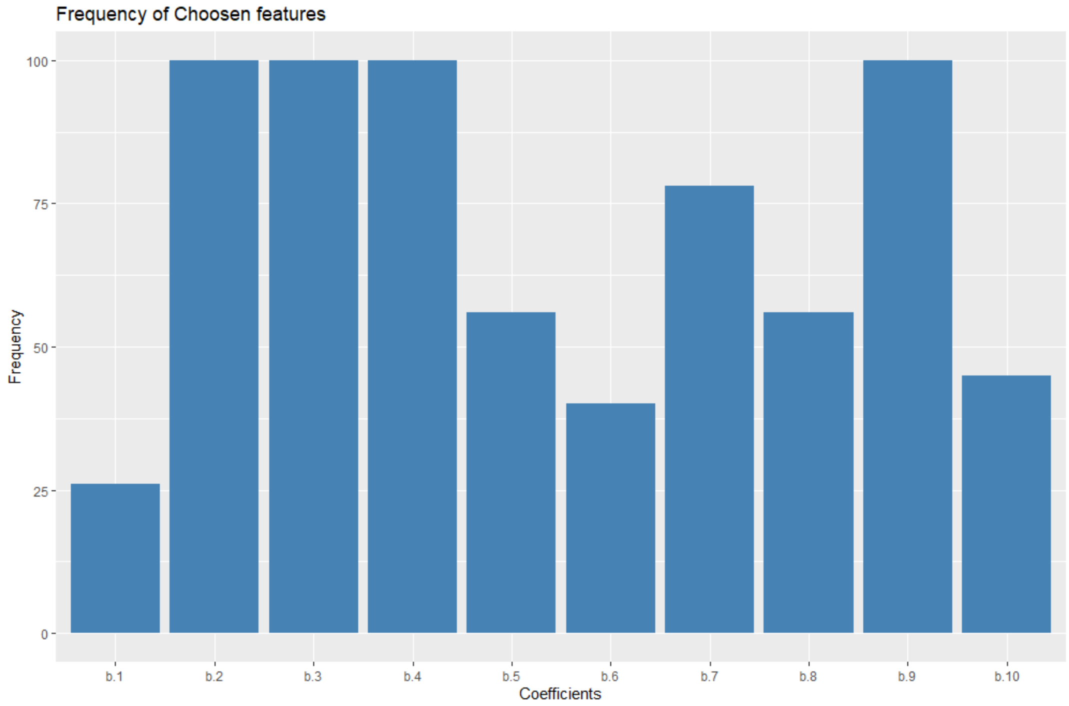

Figure 5.8: Chosen features over 100 repetitions Bayesian Lasso sparse

Figure 5.8 illustrates a Bayesian Lasso that is also sparse.

5.2 RVM, RVM$_{BLS}$ and RVM$_{BLSX}$

In this section we will test and compare the new sparse learning methods introduced in chapter 4. We will mainly look at the sparsity of each model but also as well look at prediction power in certain datasets. The following table compares the prediction performance of the models mentioned in chapter 4.

Table 5.2: RMSE for methods used in Chapter 4

Dataset Method	Boston Housing	Diabetes	Friedman 1 Data
RVM	6.127	61.242	2.239
RVM$_{BLS}$	4.587	51.251	2.218
RVM$_{BLSX}$	4.641	50.2752	2.222

Table 5.2 shows the RMSE for the three methods. Already we can notice that the RVM is

getting outperformed by our newer models. The following section will illustrate the sparsity of the models RVM$_{BLS}$ and RVM$_{BLSX}$.

Diabetes dataset

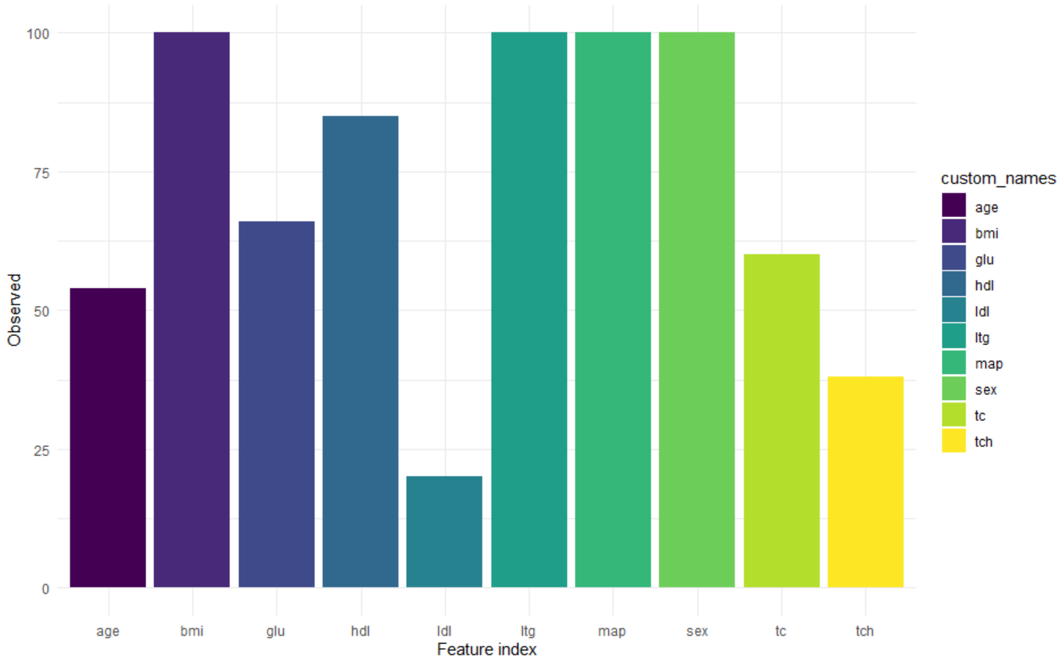

Figure 5.9: Frequency of chosen features in RVM$_{BLS}$ in the Diabetes Dataset

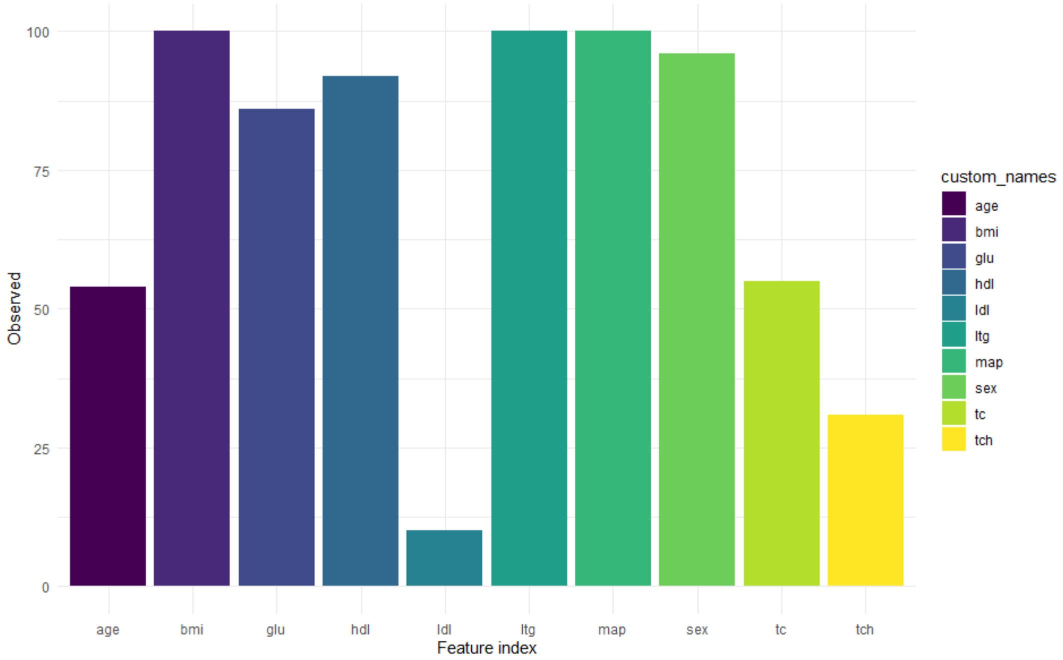

Figure 5.10: Frequency of chosen features in RVM$_{BLSX}$ in the Diabetes Dataset

In Figures 5.9 and 5.10 the analysis reveals that on average, both models tend not to select all available features. Specifically, the variables age and ldl exhibit a high frequency of non-selection within the models. This observation of sparsity is noteworthy as it indicates that the models prioritize a subset of features while disregarding others. The consistent exclusion of age and ldl suggests that these variables may have limited impact or contribute little to the predictive performance of the models.

Boston Housing dataset

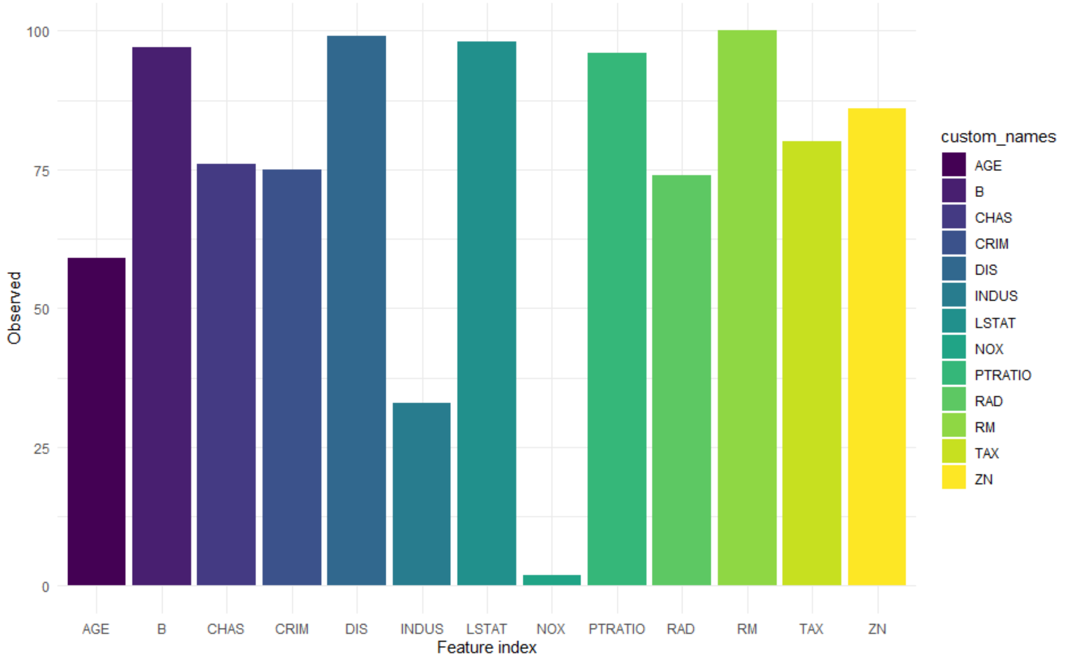

Figure 5.11: Frequency of chosen features in RVM$_{BLS}$ for Boston Dataset

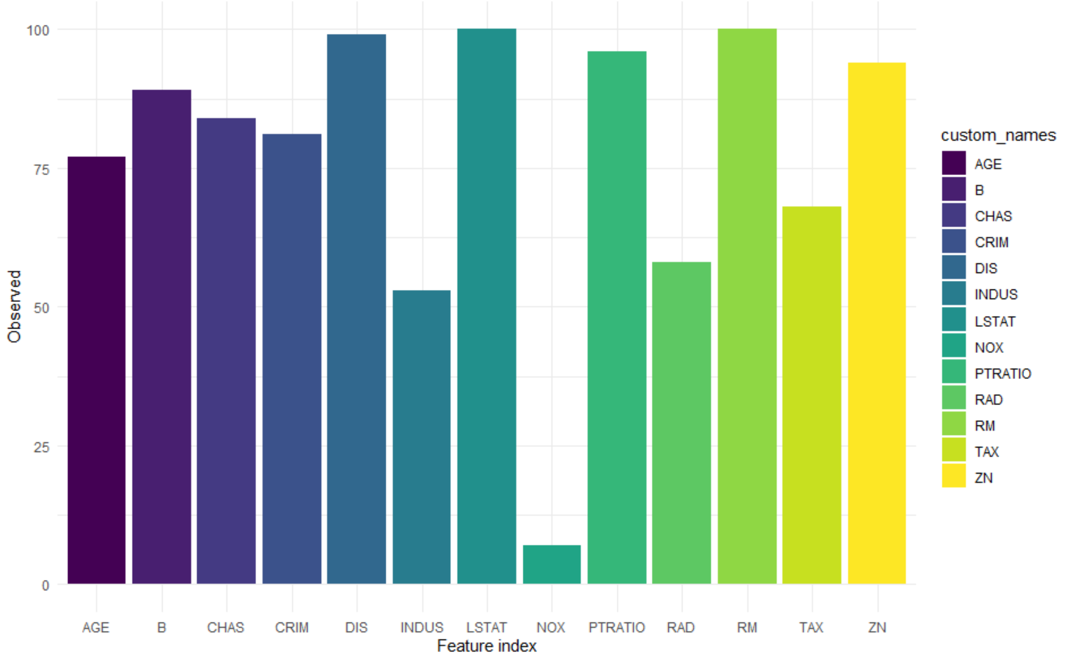

Figure 5.12: Frequency of chosen features in RVM$_{BLSX}$ for Boston Dataset

Previous research on housing markets has suggested that the concentration of nitric oxide (nox) should play a significant role in understanding housing dynamics. However, in our models, the inclusion of the nox feature seems to have an opposite effect, which goes against previous research [Harrison and Rubinfeld, 1978]. While the exact reasons for this is not yet clear, it is essential to acknowledge that machine learning models are not infallible and may encounter challenges in capturing the complexities of certain datasets. For more than 80% of the samples, the selected models do not incorporate all available features, indicating that the models tend to identify and utilize only a subset of the available predictors. RVM$_{BLSX}$ values nox slightly more than RVM$_{BLS}$ over the repetitions and excludes the RAD and TAX variable a bit more often.

Friedman #1 dataset

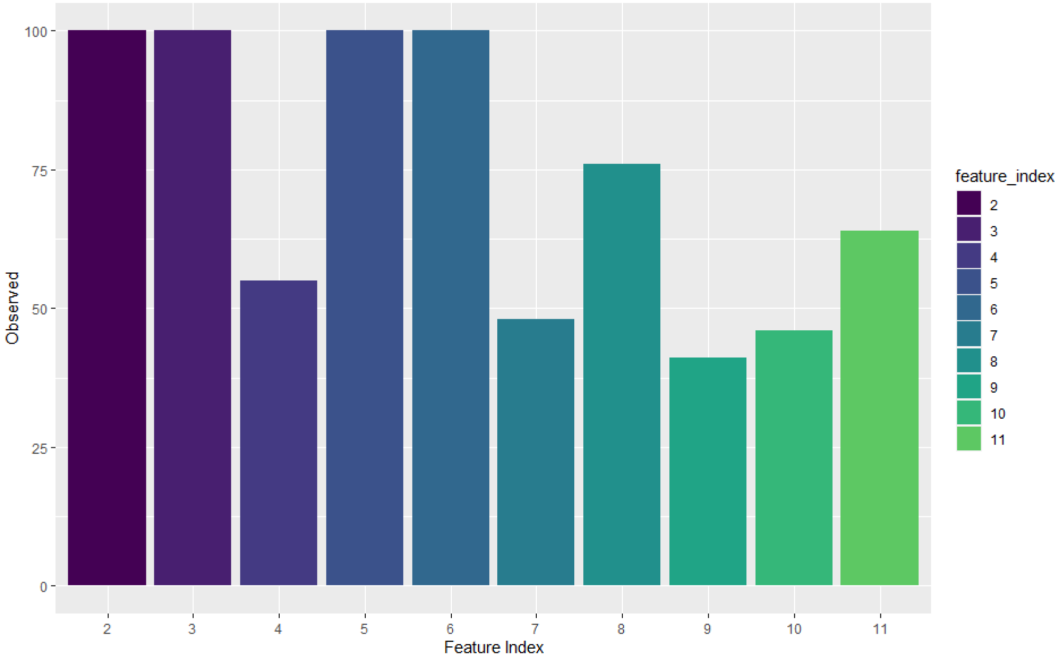

Figure 5.13: Frequency of chosen features in RVM$_{BLS}$ for Friedman #1 Data

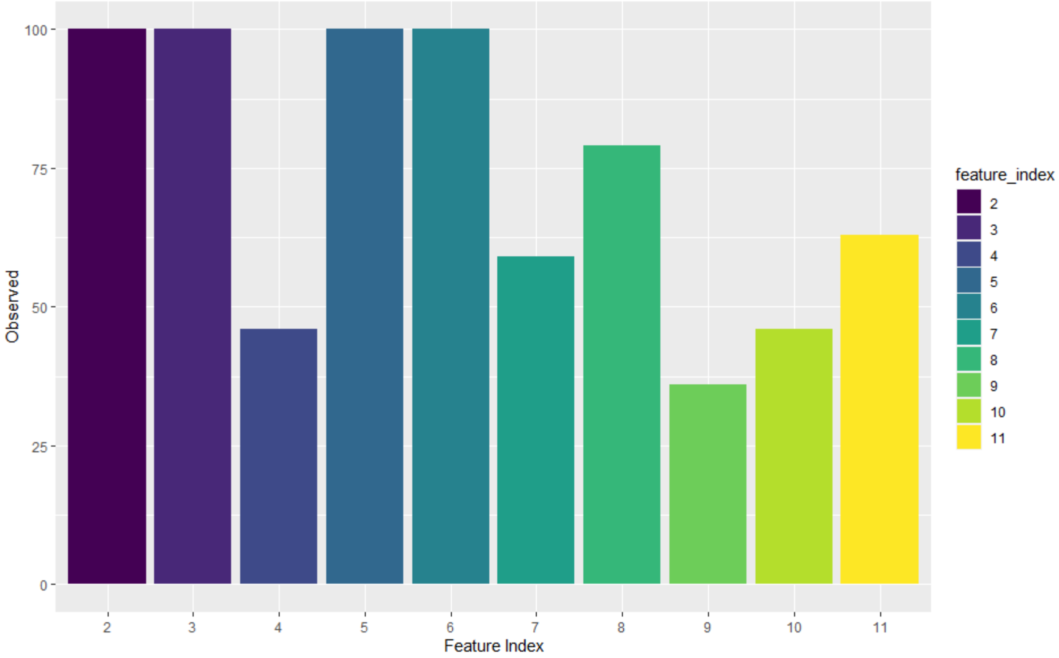

Figure 5.14: Frequency of chosen features in RVM$_{BLSX}$ for Friedman #1 Data

The figures 5.13 and 5.14 reveal a consistent pattern. The algorithms consistently select at least 4 out of the 5 columns containing relevant information from the dataset, capturing the essential features with a 100% selection rate. However, when it comes to the noise columns, the algorithms exhibit a lower selection rate, indicating their ability to differentiate between relevant features and irrelevant noise. Notably, the RVM$_{BLSX}$ algorithm demonstrates a tendency to select fewer features over time, as indicated by the slightly lower selection rate for the third and fourth columns. This trend is also observable in the selection of noise columns, suggesting that the algorithm recognizes the minimal impact these columns have on the predictions and tends to exclude them. This trade-off between feature selection and noise exclusion is a desired outcome in the analysis of the Friedman dataset. The ultimate goal is for the algorithms to assign less importance to the last five columns, effectively ignoring them due to their minimal effect on the predictions[Friedman, 1991].

5.3 Variance within the model

While experimenting with my results i noticed however how dependent the results on the initialized σ^2. In [Tipping, 2001] suggests a value of $\sigma^2 = \text{var}(\mathbf{y}) * 0.01$, while [Babacan et al., 2010] suggests a value of $\text{var}(\mathbf{y}) * 0.1$.

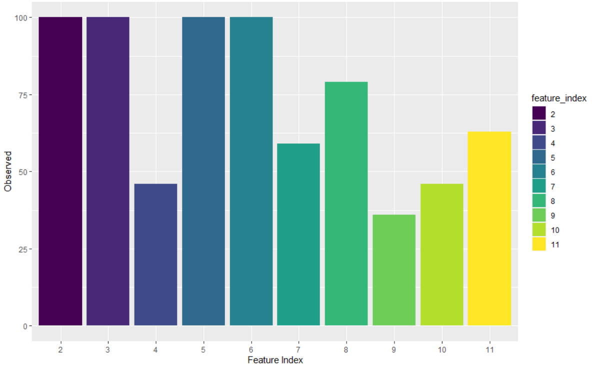

Figure 5.15: Frequency of chosen features in RVM$_{BLSX}$ with initialized as $\sigma^2 = var(\mathbf{y}) \times 0.1$

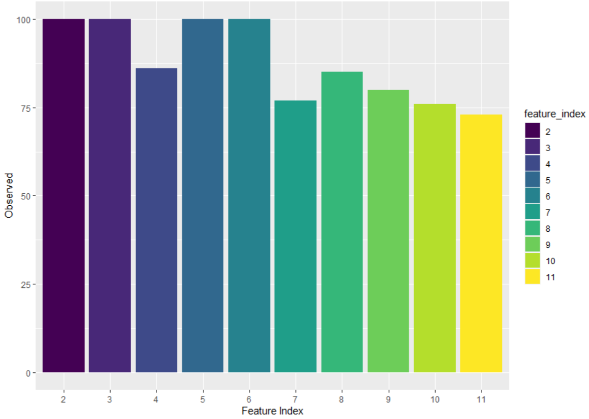

Figure 5.16: Frequency of chosen features in RVM$_{BLSX}$ with initialized $\sigma^2 = var(\mathbf{y}) \times 0.01$

The figure above show the difference in the same model with different values for σ^2. When σ^2 is initialized with $\text{var}(\mathbf{y}) \times 0.01$ it becomes less sparse. It still gives sparse samples but for the majority of repetitions it includes the majority of the features. This indicates that the initialized σ^2 might need to be estimated within the model.

Chapter 6

Discussion and Further Work

The aim of this thesis is to get an in depth look at different techniques for supervised learning methods within the Bayesian Framework that could automatically or manually achieve variable selection, and show how each model performs on empirical examples. In statistical learning, there is no best model that fits for all examples as shown, however the models represented do have their advantages in certain scenarios over other models. We present different supervised learning models where Ridge and Bayesian Lasso can not achieve sparsity automatically, while Lasso can. We also illustrated the model estimation in frequentist and Bayesian perspectives. The odd thing out is the predictive power of Bayesian Lasso, as it seems to be worse then both Lasso and Ridge, this might have to do with initialization of the model in R, or it just happens to be the case for these datasets.

We have successfully developed and implemented a new method RVM_{BLSX} which seems to outperform (on these examples) the RVM with better prediction and sparsity which is a desired result. However, RVM_{BLSX} is small modification on the original RVM_{BLS}, as our result in terms of prediction seem to stay in the same area as RVM_{BLS}. This might be for different reasons as for different datasets it might work better or worse, but as of these three examples illustrated in the paper, RVM_{BLSX} cannot be deemed as a significant improvement in prediction. If we compare the RVM_{BLS} and RVM_{BLSX} in terms of sparsity, RVM_{BLSX} seems marginally sparser on these three datasets. This might imply that it produces more sparse models in general, but that is hard to determine with three examples.

For further work, I would suggest to estimate σ^2 from a Gibbs sampler within the algorithm and extend RVM_{BLSX} to classification.

Appendix A

Appenix

A.1 Hierarchical Model And Gibbs sampler

The Gibbs Sampler is a Markov chain Monte Carlo algorithm employed to generate a sequence of observations that approximate a specific multivariate probability distribution when direct sampling from the model becomes challenging [Park and Casella, 2008].

For the Bayesian Lasso, we utilize a Gibbs sampler that leverages the representation of the Laplace distribution as a scale mixture of normals with an exponential mixing density [Park and Casella, 2008]).

$$\begin{aligned} \frac{a}{2}e^{-a|z|} &= \int_0^\infty \frac{1}{\sqrt{2\pi s}}e^{\frac{-z^2}{2s}}\frac{a^2}{2}e^{-\frac{a^2 s}{2}}ds, \\ & \qquad\qquad\qquad a > 0 \end{aligned} \tag{A.1}$$

In ([Andrews and Mallows, 1974],[Park and Casella, 2008]) it is suggested that the hierarchical representation of the model is:

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

By integrating out $\tau_1^2,...,\tau_p^2$, we can obtain the desired form for the conditional prior on

β. Additionally, if we use inverse-gamma priors for σ^2, conjugacy is preserved. Since the columns of $\mathbf{X}$ are centered, integrating out μ from the joint posterior under the independent flat prior becomes straightforward. We often find that μ is not particularly informative, so marginalizing it out simplifies the computations and improves speed. Importantly, marginalizing μ does not affect conjugacy. To sample from the conditional distribution of β, σ^2, and $\tau_1^2, ..., \tau_p^2$, we need to consider their dependencies on the vector $\tilde{y}$.

The full conditional for β is just the multivariate normal with mean $\mathbf{A}^{-1}\mathbf{X}^T\tilde{\mathbf{y}}$ and variance $\sigma^2\mathbf{A}^-1$ where $\mathbf{A} = \mathbf{X^TX} + \mathbf{D}_\tau^\mathbf{1}$. And the full conditional for σ^2 is a inverse-gamma with shape parameter:

$$\frac{(n-1)}{2} + \frac{p}{2} \tag{A.3}$$

And scale parameter:

$$\frac{(\tilde{\mathbf{y}} - \mathbf{X}\beta)^\mathbf{T}(\tilde{\mathbf{y}} - \mathbf{X}\beta)}{\mathbf{2} + \beta^\mathbf{T}\mathbf{D}_\tau^-\mathbf{1}\beta/\mathbf{2}} \tag{A.4}$$

Where $\tau_1^2, ..., \tau_p^2$ are conditionally independent with $\frac{1}{\tau_j^2}$ conditionally inverse-Gaussian with parameters:

$$\mu = \sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}} \qquad \lambda = \lambda^2 \tag{A.5}$$

The density of the inverse-Gaussian density is given by:

$$f(x) = \sqrt{\frac{\lambda'}{2\pi}x^{-3/2}} exp\Big\{-\frac{\lambda'(x-\mu')^2}{2(\mu')^2 x}\Big\} \qquad x > 0 \tag{A.6}$$

These full conditionals form the basis of an efficient Gibbs sampler with block updating of β and $(\tau_1^2, ..., \tau_p^2)$.

A.1.1 Implementation of the Gibbs Sampler

We use the inverse gamma prior distribution om σ^2

$$\begin{aligned}\pi(\sigma^2) &= \frac{\gamma^a}{\Gamma(a)}(\sigma^2)^{-a-1}e^{\frac{\gamma}{\sigma^2}} \\ \sigma^2 &> 0 \\ (a > \gamma > 0)\end{aligned} \tag{A.7}$$

The conjugate priors can vary from models but we stick to one[Park and Casella, 2008]. Here we assume a independent, flat prior on μ with the hierarchy of:

$$\begin{aligned}\mathbf{y}|\mu,\mathbf{X},\beta,\sigma^2 &\sim \mathcal{N}_n(\mu\mathbf{1}_n+\mathbf{X}\beta,\sigma^2\mathbf{I}_n) \\ \beta|\sigma^2,\tau_1^2,...,\tau_p^2 &\sim n_p(0_p,\sigma^2\mathbf{D}_\tau), \\ \mathbf{D}_\tau &= diag(\tau_1^2,...,\tau_p^2), \\ \sigma^2,\tau_1^2,...,\tau_p^2 &\sim \pi(\sigma^2)d\sigma^2\prod_{j=1}^{p}\frac{\lambda^2}{2}e^{\frac{-\lambda^2\tau_j^2}{2}}d\tau_j^2 \\ \sigma^2,\tau_1^2,...,\tau_p^2 &> 0\end{aligned} \tag{A.8}$$

By integrating out all the hyperparameters in the second equation above, we get the marginal prior for β as the following conditional priors:

$$\pi(\beta|\sigma^2) = \prod_{j}^{p}\frac{\lambda}{(2\sqrt{\sigma^2})}e^{(-\lambda|\beta_j|\sqrt{\sigma^2})} \tag{A.9}$$

making the joint density:

$$\begin{aligned}f(\mathbf{y}|\mu,\beta,\sigma^2)\pi(\sigma^2)\pi(\mu)\prod_{j=1}^{p}\pi(\beta_j|\tau_j^2,\sigma^2)\pi(\tau_j^2) &= \\ \frac{1}{(2\pi\sigma^2)^{n/2}}\exp(\frac{1}{2\sigma^2})(\mathbf{y}-\mathbf{1}_n-\mathbf{X}\beta)^T(\mathbf{y}-\mu\mathbf{1}_n-\mathbf{X}\beta) \\ \times\frac{\gamma^a}{\Gamma(a)}(\sigma^2)^{-a-1}e^{\gamma/\sigma^2}\prod_{j=1}^{p}\frac{1}{(2\pi\sigma^2\tau_j^2)^{1/2}}\exp(\frac{1}{(2\sigma^2\tau_j^2)^{1/2}}\beta_j^2)\frac{\lambda^2}{2}e^{-\lambda^2\tau_j^2/2}\end{aligned} \tag{A.10}$$

If we define $\bar{y}$ as the average of the elements of $\mathbf{y}$ we get:

$$\begin{aligned}&(\mathbf{y}-\mu\mathbf{1}_n-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{1}_n-\mathbf{X}\beta) \\ &= (\bar{y}-\mu\mathbf{1}_n)^T(\bar{y}\mathbf{1}_n-\mu\mathbf{1}_n)+(\tilde{\mathbf{y}}-\mathbf{X}\beta)^T(\tilde{\mathbf{y}}-\mathbf{X}\beta) \\ &= n(\bar{y}-\mu)^2+(\tilde{y}-\mathbf{X}\beta)\end{aligned} \tag{A.11}$$

Since the columns of $\mathbf{X}$ are standardised, the full conditional of μ is normal with the mean of $\bar{y}$ and variance $\frac{\sigma^2}{n}$. As done previously we can integrate μ out, leaves us with a joint density which is only marginal over μ proportional to:

$$\frac{1}{(\sigma^2)^{(n-1)/2}}\exp(-\frac{1}{2\sigma^2})(\tilde{\mathbf{y}}-\mathbf{X}\beta)^T(\tilde{\mathbf{y}}-\mathbf{X}\beta)(\sigma^2)^{-a-1}e^{\gamma/\sigma^2}\prod_{j=1}^{p}\frac{1}{(\sigma^2\tau_j^2)^{\frac{1}{2}}}e^{\frac{1}{2\sigma^2\tau_j^2}}\beta_j^2e^{-\lambda^2\tau_j^2/2} \tag{A.12}$$

The expression above depends on $\mathbf{y}$ only through $\tilde{\mathbf{y}}$. The conjugacy of the other parameters remains unaffected and thus making it easy to form a Gibbs sampler for β,σ^2 and $(\tau_1^p,...,\tau_p^2)$ based on this density [Park and Casella, 2008]. For β the full conditional is multivariate normal and the exponent terms involving β is:

$$-\frac{1}{2\sigma^2}(\tilde{\mathbf{y}}-\mathbf{X}\beta)^T(\tilde{\mathbf{y}}-\mathbf{X}\beta)--\frac{1}{2\sigma^2}\beta^T(\mathbf{X}^T+\mathbf{X}+\mathbf{D}_\tau^{-1})\beta-2\tilde{\mathbf{y}}^T(\mathbf{X}\beta+\tilde{y}^T\tilde{y}) \tag{A.13}$$

If we let $\mathbf{A}=\mathbf{X}^T\mathbf{X}+\mathbf{D}_\tau^{-1}$ and do a square transformations, the above equation turns into:

$$\beta^T\mathbf{A}\beta-2\tilde{y}^T\mathbf{X}\beta+\tilde{y}^T\tilde{y}=(\beta-\mathbf{A}^{-1}\mathbf{X}^T\tilde{y})^T\mathbf{A}(\beta-\mathbf{A}^{-1}\mathbf{X}^T\tilde{y})+\tilde{y}^T(\mathbf{I}_n-\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T)\tilde{y} \tag{A.14}$$

This makes β conditionally multivariate normal with mean $\mathbf{A}^{-1}\mathbf{X}^T\tilde{y}$ and variance $\sigma^2\mathbf{A}^{-1}$ [Park and Casella, 2008].

A.2 Unimodality under prior π

The joint posterior $\pi(\beta,\sigma^2\tilde{1})$ with a β and a $\sigma^2>0$ under the prior

$$\pi(\beta|\sigma^2)=\pi(\sigma^2)\prod_{j}^{p}\frac{\lambda}{(2\sqrt{\sigma^2})}e^{-\lambda|\beta_j|\sqrt{\sigma^2})} \tag{A.15}$$

When considering the statistical context, a unimodal distribution refers to a probability distribution that exhibits a single peak. In this context, the term "mode" refers to any peak of the distribution, not limited to the specific definition of mode in traditional statistics. For typical choices of $\pi(\sigma^2)$ (the prior distribution for σ^2) and for any choice of $\lambda\geq 0$, it can be observed that the conditional distribution of σ^2 is unimodal. In the context of the upper level set, denoted as $c>0$, it can be expressed as follows:

$$\{(\beta,\sigma^2)|\pi(\beta,\sigma^2|\tilde{y}) > c, \sigma^2 > 0\} \tag{A.16}$$

The upper level set is connected, and the log posterior is:

$$ln(\pi(\sigma^2)) - \frac{n+p-1}{2} ln(\sigma^2) - \frac{1}{2\sigma^2}\tilde{\mathbf{y}} - \mathbf{X}\beta_2^2 - \lambda\phi \tag{A.17}$$

The coordinate transformation when we exclude β and σ^2 is then defined by:

$$\begin{aligned} \phi &\Leftrightarrow \frac{\beta}{\sqrt{\sigma^2}} \\ p &\Leftrightarrow \frac{1}{\sqrt{\sigma^2}} \end{aligned} \tag{A.18}$$

The conditional distribution of σ^2 is continuous and exhibits a continuous inverse when $0 < \sigma^2 < \infty$. This implies that there is a one-to-one mapping between the values of σ^2 and the corresponding probabilities, allowing for a smooth transformation between the two.Furthermore, unimodality in the original coordinates is equivalent to unimodality in these transformed coordinates. Therefore, if the conditional distribution of σ^2 is unimodal in the original coordinates, it remains unimodal after the transformation. To express the equation for the upper level set, denoted as $c > 0$, we can proceed as follows:

$$ln(\pi(\sigma^2)) + (n+p-1)ln(p) - \frac{1}{2}||p\tilde{y} - \mathbf{X}\phi||_2^2 - \lambda||\phi||_1 \tag{A.19}$$

The 2. and 4. terms in the equation above is concave in (ϕ, p) and the 3. term is a concave quadratic in (ϕ, p) making the equation concave and this the posterior is unimodal, as long as $ln(\pi(\frac{1}{p}))$ is concave. This can result in instances where σ^2 has the scale invariant prior $\frac{1}{\sigma^2}$ or any inverse-gamma prior.

A.3 Basis function

This section is strictly from [Tipping and Faul, 2003] explaining the mathematical operations behind how we operate with the basis functions within all extensions of the RVM.

Adding a new basis function

$$
\begin{aligned}
2\Delta\mathcal{L} &= \frac{Q_i^2 - S_i}{S_i} + \log\frac{S_i}{Q_i^2}, \\
\tilde{\Sigma} &= \begin{bmatrix} \Sigma + \beta^2\Sigma_{ii}\Sigma\Phi^T\phi_i\phi_i^t\Phi\Sigma & -\beta^2\Sigma_{ii}\Sigma\Phi^T\phi_i \\ \beta^2\Sigma_{ii}(\Sigma\Phi^T\phi_i)^T & \Sigma_{ii} \end{bmatrix} \\
\tilde{\mu} &= \begin{bmatrix} \mu - \mu_i\beta\Sigma\Phi^T\phi_i \\ \mu_i \end{bmatrix} \\
\tilde{S}_m &= S_m - \Sigma_{ii}(\beta\phi_m^T\mathbf{e}_i)^2 \\
\tilde{Q}_m &= Q_m - \mu_i(\beta\phi_m^T\mathbf{e}_i)
\end{aligned}
\tag{A.20}
$$

where $\Sigma_{ii} = (\alpha_i + S_i)^{-1}$ and $\mu_i = \Sigma_{ii}Q_i$. Where we define $\mathbf{e}_i \triangleq \phi_i - \beta\Phi\Sigma\Phi^T\phi_i$.

Re-Estimating a Basis Function

Define k_j as $k_j \triangleq (\Sigma_{jj} + (\tilde{\alpha}_i - \alpha_i)^{-1})^{-1}$ and Σ_j as the j-th column of Σ.

$$
\begin{aligned}
2\Delta\mathcal{L} &= \frac{Q_i^2}{S_i + [\tilde{\alpha}_i^{-1} - \alpha_i^{-1}]^{-1}} \\
\tilde{\Sigma} &= \Sigma - k_j\Sigma_j\Sigma_j^T \\
\tilde{\mu} &= \mu - k_j\mu_j\Sigma_j \\
\tilde{S}_m &= S_m + k_j(\beta\Sigma_j^T\Phi^T\phi_m)^2 \\
\tilde{Q}_m &= Q_m + k_j\mu_j(\beta\Sigma_j^T\Phi^T\phi_m)
\end{aligned}
\tag{A.21}
$$

Deleting a Basis function

$$
\begin{aligned}
2\Delta\mathcal{L} &= \frac{Q_i^2}{S_i - \alpha_i} - \log(1 - \frac{S_i}{\alpha_i}) \\
\tilde{\Sigma} &= \Sigma - \frac{1}{\Sigma_{jj}}\Sigma_j\Sigma_j^T \\
\tilde{\mu} &= \mu - \frac{\mu_j}{\Sigma_{jj}}\Sigma_j \\
\tilde{S}_m &= S_m + \frac{1}{\Sigma_{jj}}(\beta\Sigma_j^T\Phi^T\phi_m)^2 \\
\tilde{Q}_m &= Q_m + \frac{\mu_j}{\Sigma_{jj}}(\beta\Sigma_j^T\Phi^T\phi_m)
\end{aligned}
\tag{A.22}
$$

Bibliography

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(1):99–102, 1974. ISSN 00359246. URL http://www.jstor.org/stable/2984774. 4.1.4, A.1

Babacan, S. Derin, Molina Rafael, and Aggelos K Katsaggelos. Bayesian compressive sensing using laplace priors. *IEEE Transactions on Image Processing*, 19, 2010. ISSN 10577149. doi: 10.1109/TIP.2009.2032894. 4.1.1, 4.1.2, 4.1.2, 4.1.2, 5.3

R. Bellman. Dynamic programming. 153:34–37, 01 1966. 2.1

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130401. URL https://doi.org/10.1145/130385.130401. 2.1

Mohammad Chowdhury and Tanvir Turin. Variable selection strategies and its importance in clinical prediction modelling. *Family Medicine and Community Health*, 8:e000262, 02 2020. doi: 10.1136/fmch-2019-000262. 2.1

Pedro Domingos. The limitations of machine learning. *Communications of the ACM*, 55(1): 55–65, 2012. doi: 10.1145/2063176.2063195. URL https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf.

G. Geoffrey Vining Douglas C. Montgomery, Elizabeth A. Peck. Introduction to linear regression analysis, fifth edition. *International Statistical Review*, 81:69–80, 08 2013. doi: 10.1111/insr.12020_10. 3.1, 3.1

Bradley Efron, Trevor Hastie, Iain Johnstone, and Rob Tibshirani. Least angle regression (with discussions). *The Annals of Statistics*, 32, 01 2004. 5

Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1): 1–67, 1991. ISSN 00905364. URL http://www.jstor.org/stable/2241837. 5, 5.2

David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. ISSN 0095-0696. doi: https://doi.org/10.1016/0095-0696(78)90006-2. URL https://www.sciencedirect.com/science/article/pii/0095069678900062. 5, 5.2

Invild Margrethe Helgøy and Yushu Li. A bayesian lasso based sparse learning model. 1, 2023. To be published. 1, 2.1, 4.1, 4.1.1, 4.1.1, 4.1.1, 4.1.2, 4.1.2, 4.1.3

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970a. 3.1, 3.3, 3.3

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970b. 3.1, 3.3, 3.3

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 1st edition, 2013a. 3.4

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 1st edition, 2013b.

Tahira Jamil and Cajo Ter Braak. Selection properties of type ii maximum likelihood (empirical bayes) in linear models with individual variance components for predictors. *Pattern Recognition Letters*, 33:1205–1212, 07 2012. doi: 10.1016/j.patrec.2012.01.004. 3.6.1, 3.6.1

L.E. Melkumova and S.Ya Shatskikh. Comparing ridge and lasso estimators for data analysis. *Procedia Engineering*, 201, 2017. ISSN 18777058. doi: 10.1016/j.proeng.2017.09.615. 3.3, 3.3

Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103, 2008. ISSN 01621459. doi: 10.1198/016214508000000337. 1, 3.5, 3.5, 3.5, 3.5, 4.1, 4.2, 4.2, 4.2, 4.2, A.1, A.1, A.1.1, A.1.1, A.1.1

Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 07 2001. doi: 10.1162/089976601750264965. 2.1

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. 3.1, 3.2

Michael Tipping. Sparsebayes software. *arXiv preprint arXiv:1608.06331*, 2016.

Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1, 2001. ISSN 15324435. 1, 2.1, 2.2, 3.6, 3.6, 3.6, 3.6, 4.1.2, 4.1.4, 5.3

Michael E. Tipping and Anita C. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 276–283. PMLR, 03–06 Jan 2003. URL https://proceedings.mlr.press/r4/tipping03a.html. 3.6.1, 3.6.1, 4.1.2, A.3

Vladimir Vapnik, Steven E. Golowich, and Alex J. Smola. Support vector method for function approximation, regression estimation and signal processing. In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9 — Proceedings of the 1996 Neural Information Processing Systems Conference (NIPS 1996)*, pages 281–287, Dever, CO, USA, December 1997. MIT Press, Cambridge, MA, USA. URL http://dblp.uni-trier.de/db/conf/nips/nipsN1996.html#VapnikGS96. 2.1

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x. 5.1