Università degli Studi di Milano-Bicocca

SCUOLA DI ECONOMIA E STATISTICA
Corso di Laurea Magistrale in Scienze Statistiche ed Economiche

Tesi di laurea magistrale

Improving ridge regression via model selection and focussed fine-tuning

Candidato:
Riccardo Parviero
Matricola 751762

Relatore:
Prof.ssa Sonia Migliorati

Correlatori:
Prof. Nils Lid Hjort
Dott. Kristoffer H. Hellton

Anno Accademico 2016 - 2017

Abstract

Ridge regression is an L_2 penalized regression method that depends on a penalty parameter. Among the techniques used to fine-tune the value of this parameter, cross-validation is well established. As an alternative to cross-validation, we suggest two procedures based on the minimization of the expected estimation error and the expected prediction error of ridge regression, with the aid of suitable plug-in estimates. We demonstrate that these mean squared error expression could be used as averaged focussed information criteria. This way, it is possible to develop a model selection method based on ridge regression. To demonstrate the approach, we assessed its performance in both a simulation study and in a real data application. Both studies came to the conclusion that our method is capable of detecting the most important covariates of a given dataset and yields a prediction error comparable to cross-validation. We also present an information criterion tailored for the prediction of a specific unit, which can also be used to perform personalized covariate selection.

Contents

List of Tables

List of Figures

Preface

The process of writing this thesis was not conventional, to say the least. Coming from outside, I reached professor Nils Lid Hjort in Oslo, who brought to join us also Dr. Kristoffer H. Hellton. We explored several ideas regarding penalized regression methods and how could they be enhanced or reformed regarding the fine-tuning methods and other aspects. The topic we eventually chose aimed at creating a link between model selection and ridge regression, and it will be the main topic presented in this thesis. Even though something did not go as planned, also exploring why some parts did not work as expected was definitely part of the learning process. It has certainly been an important step in my professional and personal lives and it all started with an e-mail I sent to prof. Hjort almost a year ago, on March 10th 2017.

I want to thank professor Nils Lid Hjort and Dr. Kristoffer Hellton for having supported me throughout all the process and for having welcomed me in their research group. Their feedback was always useful and helped me through everything, and I really appreciated that.

I want also to thank my supervisor in Italy, professor Sonia Migliorati, who trusted me since the first day and backed me in my choice of taking this very important step away from my home university. Without her support, none of this would have been possible.

Chapter 1

Introduction

Two of the main challenges that statistics has are prediction and model selection. If one wants to pursue the former, the only aim is to reach the most reliable predictions possible, in terms of some performance metric. The latter framework, instead, aims at drawing the most plausible conclusions from the present data. Within this framework, one is interested in selecting the true model that explains the real mechanisms behind the data. Aiming attention solely towards future prediction performances could push the model out of the classical inference framework, for instance, a good weather forecast might perform better in predicting the temperature of the next few days even if it is not completely accurate in any other task, such as predicting the current temperature. On the other hand there are situations in which the model has primarily to perform model selection: for example, while trying to assess if the presence of an important disease could depend on some genetic expression, performing model selection would be to identify with the most precision as possible which genes are actually responsible for such disease.

These two approaches, however, do not necessarily have to be in conflict with each other. One may in fact think that a thorough model selection procedure could also have benefits regarding prediction performances. As explained earlier, the main goal of model selection is to identify the real structure behind what we want to explain, hence having a method that can discern between what is really linked with what we want to explain and what is not, is always of great help. The main goal of this thesis is to find a method that could go in both directions, starting from model selection.

1.1 Linear regression

When given a quantitative outcome that one wants to explain using other variables, a first method that can be used is linear regression. The linear regression method assumes the outcome to be a linear function of those variables with additive noise:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where the design matrix $\mathbf{X}$ contains the recorded values of p covariates for each of the n sampled units and the vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$ contains the n values of the response variable.

The p unknown *regression parameters* are stored in the vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)|top$, and can be estimated by minimizing the square distance between the estimated function and the observed outcome, also termed residual sums of squares

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \{(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})\}.$$

The optimization problem has an enclosed form solution given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y}.$$

The resulting estimator of the regression parameters is called *ordinary least squares* estimator. Even though this approach might be simple, it has upsides for his inference abilities. When the independent error term, $\boldsymbol{\epsilon}$, is assumed to be normally distributed, it is possible to draw inference on the parameters in $\boldsymbol{\beta}$. Since the most common assumption involves the hypothesis that the error term has mean zero, the ordinary least squares estimator (or OLS estimator) is unbiased and it is also the best linear unbiased estimator, according to Gauss-Markov Theorem. Having the unbiasedness property, the ordinary least squares estimator is hence *expected* to estimate the true values of the regression coefficients.

1.2 The bias-variance trade-off

When the aim of an analysis is to predict future values of the outcome, one would want to use a method that could achieve the lowest error possible while estimating the regression parameters. If the outcome is a continuous variable, the prediction error is frequently assessed through a metric called *mean squared error*. This metric is computed using the expectation

of the square distance between the estimates and the true values of the parameters. The definition of the mean squared error of $\hat{\theta}$, estimator of the parameter, θ, is

$$\text{MSE}[\hat{\theta}] = \mathbb{E}[(\hat{\theta} - \theta)^2].$$

The manipulation of the formula gives the well-known bias-variance decomposition of the mean squared error

$$\begin{aligned}\text{MSE}[\hat{\theta}] &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta)^2], \\ &= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] + (\mathbb{E}[\hat{\theta}] - \theta)^2, \\ &= \mathbb{V}[\hat{\theta}] + \text{Bias}^2[\hat{\theta}],\end{aligned}$$

which highlights a trade-off between the squared bias and the variance of the estimator. Hence, sometimes it could be a better choice to use a method that can exploit the trade-off introducing bias into the estimates, while achieving less variability. In this setting, then, the ordinary least squares estimator has not any flexibility. Since it is an unbiased estimator by construction, it is not able to exploit the bias-variance trade-off.

One other regression method that can exploit the bias-variance trade-off is *ridge regression* [Hoerl and Kennard, 1970]. As presented earlier, the ordinary least squares estimator is obtained through an optimization problem. The ridge estimator is obtained when the same optimization is carried out by adding a penalty on the square length of the coefficients vector $\boldsymbol{\beta}$

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\text{argmin}}\{(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} \beta_i^2\}.$$

This constrained optimization problem still offer an explicit formula for the solution, given by

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\boldsymbol{y}.$$

This expression indicates that the ridge regression estimator depends on the *penalty* or *tuning* parameter, λ. We will show that by taking larger values of the tuning parameter, λ, the ridge estimator can introduce a certain amount of bias into its estimates, hence reducing its variance and achieving a lower mean squared error. The most widely used technique to assess the value of the tuning parameter, λ, is cross-validation, which tries to emulate future predictions by splitting the data into a *training* set, and a *test* set. The training set emulates present data, while the test set represents an independent sample which corresponding outcomes values are set to predict. In this thesis, we will present an

alternative technique. The attention of this method is aimed towards the estimation of the value of the tuning parameter, λ, which minimizes the expected mean squared error when all the unknown components of it are known. This value is hence called the *oracle* value of the tuning parameter, λ. We will show that the estimation of the oracle value can be accomplished by substituting the unknown quantities involved with suitable plug-in estimates.

It can be also shown that the tuning parameter, λ, permits to broaden the applications of the regression method. The computation of the ordinary least squares estimator of the regression coefficients requires the inversion of the $\mathbf{X}^\top\mathbf{X}$ matrix. In some situation this is not possible, namely when the number of covariates p is greater than the number of observations n. In addition, when $p < n$ and the data is characterized by the presence of high collinearity, the inversion of $\mathbf{X}^\top\mathbf{X}$ might still be possible, but the estimates carried out tend to be unstable. By adding a constant to the diagonal of this matrix, the ridge estimator solves both problems and hence is able to estimate the regression parameter when the number of covariates, p, exceeds the number of units, n, and to always yield stable estimates of $\boldsymbol{\beta}$.

1.3 Model selection

In some situations, however, predicting new outcomes might not to be the aim of the analysis. When the data comprises several covariates, one may suppose that not all of them is truly connected to the outcome. Finding which covariates have a true effect on the response variable is what is termed *model selection* or *variable selection*. A common procedure of doing model selection starts by dividing the p covariates into subsets and then searching for the true model through those subsets. Other than selecting the true covariates, model selection also helps to simplify interpretation of a model. This aspect is connected with the idea of *parsimony*: at times, it is better to select a smaller model, but which is certain to explain the main structure behind the response variable. Selecting a parsimonious model, eventually, could also be one step closer to the truth following the Ockham's razor reasoning, which states that the simplest hypothesis could be the most likely to be true, *caeteris paribus*.

In order to perform model selection, several methods can be used. A possible choice is to use an information criterion as the Akaike's information criterion [Akaike, 1973], also known as AIC, or the Byesian information criterion [Schwarz et al., 1978], also known as BIC. Such methods drive the model selection by penalizing the likelihood of a certain model S drawn from the list of submodels one wants to choose from, with a penalty depending to

their complexity in terms of number of parameters estimated. The model that scores the highest value of the criterion is then deemed to be chosen. Hence, the bigger the model, the more the penalty will counterbalance its likelihood value, with the aim of finding a model with enough parameters to explain the response variable while still being parsimonious. The penalty of the BIC depends also on the sample size n, hence it can be demonstrated that this criterion has the property of *consistency*, which means that it will tend to select the true model almost surely when the sample size increases and hence more information is used.

An alternative way to perform model selection is to use a type of penalized regression different from ridge regression, which is *lasso regression* [Tibshirani, 1996]. This technique imposes a different penalization on the regression coefficients, namely the L1 penalty, which is linked to the absolute values of those coefficients. The optimization problem associated with lasso is

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} |\beta_i|\}.$$

With this new penalty, the lasso optimization problem does not offer an explicit solution for the estimator of the regression coefficients. On the other hand, it is now possible to carry out sparse estimates of the vector $\boldsymbol{\beta}$. A sparse vector is characterized by the presence of null elements, hence when some regression coefficients are estimated to be exactly zero, it is possible to interpret them as non effective on the outcome. As the value of the tuning parameter, λ, grows, the estimates produced by the lasso method are sparser and sparser, hence removing from model more and more variables. Ridge regression lacks entirely this property, as its estimates are *shrinked* towards zero, but none of them will reach that value unless we let the tuning parameter λ tend to infinity.

These model selection techniques have a common trait: they will select the model which has to serve a general purpose. Such purpose can be identified in the AIC and BIC criteria with the model that yields the best overall fit, whereas lasso will select the model that is considered to yield the best predictions, in terms of prediction mean squared error, when the cross-validation technique is used to choose the value of its tuning parameter. When a particular quantity or parameter is considered to have a larger importance a priori, the focussed information criterion [Claeskens and Hjort, 2008] helps by selecting the model that yields the best estimate for this parameter of interest, termed focus. This criterion is able to perform this type of selection by minimizing a risk function associated to the focus, μ, over every considered submodel S

$$\text{FIC}(\hat{\mu}_S) = \text{MSE}[\hat{\mu}_S] = \mathbb{E}[(\hat{\mu}_S - \mu_{true})^2].$$

This approach permits also to build averaged focussed criteria, when this criterion is considered over multiple parameters. We will show that this way it is possible to introduce model selection into the ridge regression framework. The task will be accomplished through the minimization of the expected error of the ridge estimator both over a list of possible models and over its tuning parameter, λ.

The aim of this thesis is to present a method that can implement the ability of performing model selection into ridge regression. In Chapter 2, ridge regression is presented in depth, showing its most important properties and theoretical results. Then several techniques for estimating the tuning parameter, λ, are shown, comprising the technique of expected error minimization. At the end of the chapter a simulation study is presented, in which we assessed the ability of the expected error minimization technique to estimate the oracle value of the tuning parameter. Chapter 3 starts with an overview on the most widely used model selection methods, then proceeds to present how an averaged focussed approach could be use to implement variable selection into the ridge regression framework. Also this chapter ends with a simulation study in which we assessed the ability of such method of finding the true model and a simple application on real data, in order to compare our technique with the aforementioned ones.

Chapter 2

Ridge regression

In this chapter we will present the ridge regression method [Hoerl and Kennard, 1970], as an extension of ordinary least squares regression, and several techniques used to select the value of the tuning parameter λ. Before showing the main properties of the ridge estimator, the ordinary least squares estimator is first presented.

2.1 From ordinary least squares to ridge regression

Consider a setting in which n units are sampled, and for all of them are present informations about p covariates and the value of a response variable y. If one wants to use those covariates to explain the response variable, a linear relationship is given by the model

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i \quad \text{for} \quad i = 1, \dots, n.$$

where the vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are fixed covariate vectors, with each one of them being p-dimensional as the unknown coefficient vector $\boldsymbol{\beta}$. The terms indicated with ϵ_i are the unobserved random error terms, which are assumed to be drawn from $\mathcal{N}(0, \sigma^2)$ and to be independent, hence $\text{Cov}[\epsilon_i, \epsilon_j] = 0$ for $i \neq j$. It is also possible to use matrix notation, by denoting the response variable with the vector $\boldsymbol{y}$, the error terms in $\boldsymbol{\epsilon}$ and the covariate vectors into the $n \times p$ design matrix $\mathbf{X}$ to yield

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon}$ is drawn from a multivariate normal, hence $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_{n\times 1}, \sigma^2 \mathbf{I}_n)$.

The aim of the ordinary least squares method is then to estimate the unknown vector $\boldsymbol{\beta}$ by minimizing the residual sum of squares. Being the residual sum of squares defined as

the square length of the distance between the fitted values, $\mathbf{X}\boldsymbol{\beta}$, and the real values of the response variable, $\boldsymbol{y}$, the ordinary least squares estimator is

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}}\{(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})\}.$$

If we assume that the design matrix has full rank, $\mathbf{X}^\top\mathbf{X}$ is invertible. This minimization, then, has a unique closed form solution in

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}. \tag{2.1}$$

If the aim is also to draw inference from this model, the assumptions made regarding the error term $\boldsymbol{\epsilon}$ allow to demonstrate that the ordinary least squares estimator is unbiased

$$\mathbb{E}[\hat{\boldsymbol{\beta}}] = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\,\mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}] = \boldsymbol{\beta}.$$

and hence it is expected to estimate the true values of the regression coefficients. Starting from the same assumptions, it is also possible to deduce that also the response variable, $\boldsymbol{y}$, follows a multivariate normal distribution with expected value

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}] = \mathbf{X}\boldsymbol{\beta},$$

and variance

$$\mathbb{E}[(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])^\top] = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] = \sigma^2\mathbf{I}_n.$$

Sometimes, however, it is not possible to estimate the regression parameters through an ordinary least squares approach. The matrix $\mathbf{X}^\top\mathbf{X}$ could not have full rank in some frameworks and so it would not be invertible. This type of problem is encountered when certain covariates contained in $\mathbf{X}$ suffer from perfect collinearity or high collinearity with other covariates. In the former case it would mean that the covariates are perfect linear combinations of other ones, the latter is encountered when there is high correlation between variables. The problem can be avoided by adding a positive quantity to the diagonal of $\mathbf{X}^\top\mathbf{X}$. This idea is the core of the ridge estimator, which stems from the same optimization problem of the ordinary least squares but with a modification. By introducing into the problem a penalization associated to the square norm of the regression coefficients' vector, the ridge estimator of $\boldsymbol{\beta}$ is the minimand of:

$$\hat{\boldsymbol{\beta}}_\lambda = \operatorname*{argmin}_{\boldsymbol{\beta}}\{(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\sum_{i=1}^{p}\beta_i^2\}.$$

This constrained optimization problem naturally contains the adjustment presented earlier to make the $\mathbf{X}^\top\mathbf{X}$ matrix of full rank. In fact, the parameter λ is referred to as *penalty parameter* because controls the so called *ridge penalization* term, which is related to the size of the regression coefficients. A larger value of λ would penalize non-zero coefficients resulting in which is called a *shrinkage* of the estimates. This behaviour of the ridge estimator reflects into its estimates of $\boldsymbol{\beta}$, which are characterized by having a lesser square norm of the ones produced by an ordinary least squares approach. We will show later in this work that by choosing other types of penalizations it is possible to obtain different estimators, such as the lasso, when the absolute norm is used to cover this task.

Analysing the first and second derivative of the constrained optimization problem we reach a unique solution in:

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\boldsymbol{y}, \tag{2.2}$$

which is the so-called ridge estimator. The expression of the ridge estimator can be manipulated to show its connection with the ordinary least squares estimator:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_\lambda &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y},\\ &= [\mathbf{I}_p + \lambda(\mathbf{X}^\top\mathbf{X})^{-1}]^{-1}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y},\\ &= \mathbf{W}_\lambda\hat{\boldsymbol{\beta}},\end{aligned}$$

with $\mathbf{W}_\lambda = [\mathbf{I}_p + \lambda(\mathbf{X}^\top\mathbf{X})^{-1}]^{-1}$.

As previously mentioned, when large values of the tuning parameter, λ, are used, the estimates tend to shrink towards zero. If the value chosen for this parameter is exactly zero, no shrinkage will be introduced into the estimates, hence obtaining the same estimates that would have been given by the ordinary least squares estimator. When this happens, the $\mathbf{W}_\lambda$ matrix collapses into an identity matrix of size p, therefore bringing the ridge estimator expression to coincide with the ordinary least squares one.

2.2 Expected value

The ordinary least squares is an unbiased estimator, meaning that $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$. The ridge estimator, on the other hand, has expected value:

$$\begin{aligned}\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{E}[\mathbf{W}_\lambda \hat{\boldsymbol{\beta}}], \\ &= \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbb{E}[\boldsymbol{y}], \\ &= \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}], \\ &= [\mathbf{I}_p + \lambda (\mathbf{X}^\top \mathbf{X})^{-1}]^{-1} \boldsymbol{\beta}.\end{aligned} \tag{2.3}$$

The result demonstrates that the ridge estimator is biased, when λ is strictly greater than zero. Given a $\mathbf{X}^\top \mathbf{X}$ matrix of full rank, the ridge estimator will become unbiased if and only if the value associated to λ is exactly zero, hence coinciding with the ordinary least squares estimator.

Moreover, the limit behaviour for the expected value when λ grows to infinity is

$$\lim_{\lambda \to \infty} \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] = \lim_{\lambda \to \infty} [\mathbf{I}_p + \lambda (\mathbf{X}^\top \mathbf{X})^{-1}]^{-1} \boldsymbol{\beta} = \mathbf{0}_{p \times 1}. \tag{2.4}$$

Other than connecting the ridge estimator with the OLS, the tuning porameter, λ, plays also a role in the shrinkage of the estimates. As expected from the nature of the penalty inserted into the constrained optimization framework, values of λ that tends to infinity produce an estimate of the $\boldsymbol{\beta}$ vector which will be characterized by shrinkage towards zero. It is then possible to note that, barring a limit case in which all the true values of the regression coefficients present in $\boldsymbol{\beta}$ are exactly 0, letting λ grow will inevitably introduce more bias into the estimates.

2.3 Variance

Using the same $\mathbf{W}_\lambda$ matrix as, it is possible to compute the ridge regression variance as

$$\begin{aligned}\mathbb{V}[\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{V}[\mathbf{W}_\lambda \hat{\boldsymbol{\beta}}], \\ &= \mathbf{W}_\lambda \mathbb{V}[\hat{\boldsymbol{\beta}}] \mathbf{W}_\lambda^\top, \\ &= \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{W}_\lambda^\top, \\ &= \sigma^2 (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{X} [(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1}]^\top.\end{aligned} \tag{2.5}$$

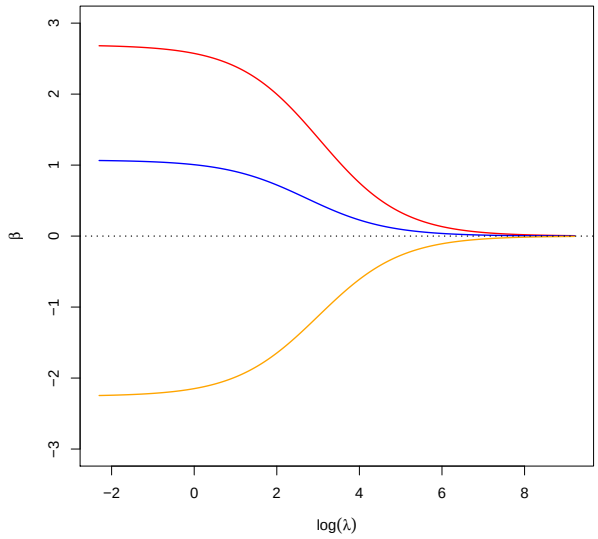


Figure 2.1: Shrinkage paths of the ridge estimator. The real coefficient vector is $\boldsymbol{\beta} = (2.5, 1.5, -3)^\top$.

Similarly to what happens to the expected value, note that the OLS is always the limit case when λ is zero, if $\mathbf{X}^\top\mathbf{X}$ is of full rank. In addition, the limit behaviour of the variance when λ tends to infinity is

$$\lim_{\lambda\to\infty} \mathbb{V}[\hat{\boldsymbol{\beta}}_\lambda] = \lim_{\lambda\to\infty} \sigma^2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\mathbf{X}[(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}]^\top = \mathbf{0}_{p\times p}. \tag{2.6}$$

A larger λ introduces more bias into the estimates but reduces the variance at the same time. The important aspect to note is that the expected value is proportional to the penalty parameter, λ, but the variance is proportional to the square of this parameter, hence the latter decreases more rapidly than the former increases and from this behaviour stems the idea that it will be always possible to find a value of the penalty parameter λ which permits to balance the two quantities in the bias-variance trade-off. For this reason, the penalty parameter, λ, may be also called *tuning parameter*: its value affects the estimates in terms of the position they occupy in the bias-variance trade-off. Hence, it has to be *fine-tuned* searching the appropriate value for every situation in order to yield the lowest mean squared error possible.

In addition, it is possible to show that when the parameter, λ, is strictly greater than zero, the ridge variance is always smaller than the variance of the ordinary least squares

method (whereas its bias is always greater than 0 in the same situation):

$$\begin{aligned}\mathbb{V}[\hat{\boldsymbol{\beta}}] - \mathbb{V}[\hat{\boldsymbol{\beta}}_\lambda] &= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \sigma^2\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top, \\ &= \sigma^2\mathbf{W}_\lambda\{\mathbf{W}_\lambda^{-1}(\mathbf{X}^\top\mathbf{X})^{-1}[\mathbf{W}_\lambda^{-1}]^\top - (\mathbf{X}^\top\mathbf{X})^{-1}\}\mathbf{W}_\lambda^\top,\end{aligned} \tag{2.7}$$

reminding that $\mathbf{W}_\lambda^{-1} = \mathbf{I}_p + \lambda(\mathbf{X}^\top\mathbf{X})^{-1}$, we obtain

$$\begin{aligned}&= \sigma^2\mathbf{W}_\lambda[(\mathbf{X}^\top\mathbf{X})^{-1} + 2\lambda(\mathbf{X}^\top\mathbf{X})^{-2} + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-3} - (\mathbf{X}^\top\mathbf{X})^{-1}]\mathbf{W}_\lambda^\top, \\ &= \sigma^2\mathbf{W}_\lambda[2\lambda(\mathbf{X}^\top\mathbf{X})^{-2} + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-3}]\mathbf{W}_\lambda^\top.\end{aligned}$$

The matrix we obtain is positive-semidefinite, hence we can conclude that $\mathbb{V}[\hat{\boldsymbol{\beta}}_\lambda] < \mathbb{V}[\hat{\boldsymbol{\beta}}]$ when λ is strictly greater than zero.

2.4 Mean squared estimation error

The quantity that is usually referred to as the mean squared error of the ridge estimator is the mean squared estimation error, or expected estimation error. In order to compute this quantity, it is used the expected length of the difference between the estimates and the real parameters contained in $\boldsymbol{\beta}$. It is always possible to compute other types of error involving the ridge estimates, one of them being the mean squared prediction error, computed by using the difference between the predicted values and the actual values of the response variable.

Starting from the estimation error, is it possible to give a formulation for the bias-variance trade-off of the ridge estimator:

$$\begin{aligned}\mathrm{MSE}[\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})^\top(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})], \\ &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] + [\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] + [\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})], \\ &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])] + \mathbb{E}[\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}]^\top\mathbb{E}[\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}] \\ &\quad + \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})] + \mathbb{E}[(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])],\end{aligned} \tag{2.8}$$

noting that $\mathbb{E}[\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda]] = \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda]$ and $\mathbb{E}[\boldsymbol{\beta}] = \boldsymbol{\beta}$,

$$\begin{aligned}&= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])] + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}) \\ &\quad + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}) + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda]),\end{aligned}$$

the last two terms are multiplied by a vector of zeroes, so we are left with the decomposition

$$= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])] + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}).$$

Knowing what form the ridge estimators and its expectation take from Equation (2.2) and Equation (2.3) respectively

$$\begin{aligned}\mathrm{MSE}[\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])] + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}) \\ &= \mathbb{E}[(\mathbf{W}_\lambda\hat{\boldsymbol{\beta}} - \mathbf{W}_\lambda\boldsymbol{\beta})^\top(\mathbf{W}_\lambda\hat{\boldsymbol{\beta}} - W_\lambda\boldsymbol{\beta})] + (\mathbf{W}_\lambda\boldsymbol{\beta} - \boldsymbol{\beta})^\top(\mathbf{W}_\lambda\boldsymbol{\beta} - \boldsymbol{\beta}) \\ &= \mathbb{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top\mathbf{W}_\lambda^\top\mathbf{W}_\lambda(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] + \boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_p)^\top(\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta} \\ &= \sigma^2\mathrm{tr}[\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top] + \boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_p)^\top(\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta}.\end{aligned} \tag{2.9}$$

To reach the final step, we have this result on the quadratic form of a normal random variable $\boldsymbol{\eta} \sim N(\boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_\eta)$

$$E(\boldsymbol{\eta}^\top\boldsymbol{\Lambda}\boldsymbol{\eta}) = \mathrm{tr}[\boldsymbol{\Lambda}\boldsymbol{\Sigma}_\eta] + \boldsymbol{\mu}_\eta^\top\boldsymbol{\Lambda}\boldsymbol{\mu}_\eta,$$

applied to $(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim N(\mathbf{0}_{n\times 1}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$.

Now, we will see that the main result that produced by this decomposition states that the mean squared error of ridge regression is always smaller than the one associated to the ordinary least squares one. In order to show such result we need first these theorem and lemma from Theobald [1974] and Farebrother [1976], respectively, which are also reported by van Wieringen [2015]:

Theorem 2.4.1.

Let $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ be two different estimators for $\boldsymbol{\theta}$ with second order moments:

$$\mathbf{M_i} = \mathbb{E}[(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta})^\top] \quad \text{with } i = 1, 2,$$

and mean squared errors:

$$\mathrm{MSE}[\hat{\boldsymbol{\theta}}_i] = \mathbb{E}[(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta})^\top\mathbf{A}(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta})] \quad \text{with } i = 1, 2,$$

where $\mathbf{A} \succeq 0$. Then, $\mathbf{M}_1 - \mathbf{M}_2 \succeq 0$ if and only if $\mathrm{MSE}[\hat{\boldsymbol{\theta}}_1] - \mathrm{MSE}[\hat{\boldsymbol{\theta}}_2] \geq 0$ for all $\mathbf{A} \succeq 0$.

Proposition 2.4.1.

If $\mathbf{A}$ is a $p \times p$ positive definite matrix and $\mathbf{b}$ is a nonzero p dimensional vector, then, for any $c \in \mathbb{R}_+$, $c\mathbf{A} - \mathbf{b}\mathbf{b}^\top \succ 0$ if and only if $\mathbf{b}^\top\mathbf{A}\mathbf{b} > c$. This result [Theobald, 1974] will show that choosing the ridge over the OLS is always a good choice in terms of mean squared error.

Theorem 2.4.2.

There always exists a $\lambda > 0$ such that $\mathrm{MSE}[\hat{\boldsymbol{\beta}}_\lambda] < \mathrm{MSE}[\hat{\boldsymbol{\beta}}]$.

Proof. The second order moment matrix of $\hat{\boldsymbol{\beta}}_\lambda$ is

$$\mathbf{M}_{\hat{\boldsymbol{\beta}}_\lambda} = \mathbb{V}[\hat{\boldsymbol{\beta}}_\lambda] + \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}]\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}]^\top,$$

Then

$$\begin{aligned}
\mathbf{M}_{\hat{\boldsymbol{\beta}}} - \mathbf{M}_{\hat{\boldsymbol{\beta}}_\lambda} &= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \sigma^2\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top - (\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta}\boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_p)^\top \\
&= \sigma^2\mathbf{W}_\lambda[2\lambda(\mathbf{X}^\top\mathbf{X})^{-2} + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-3}]\mathbf{W}_\lambda^\top \\
&\quad - \lambda^2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^\top[(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}]^\top \\
&= \sigma^2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}[2\lambda\mathbf{I}_p + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-1}][(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}]^\top \\
&\quad - \lambda^2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^\top[(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}]^\top \\
&= \lambda(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}[2\sigma^2\mathbf{I}_p + \lambda\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \lambda\boldsymbol{\beta}\boldsymbol{\beta}^\top][(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}]^\top
\end{aligned}$$

It is possible to demonstrate that this is positive definite for any λ that satisfies: $2\sigma^2(\boldsymbol{\beta}^\top\boldsymbol{\beta})^{-1} > \lambda$, meaning that $\mathbf{M}_{\hat{\boldsymbol{\beta}}} - \mathbf{M}_{\hat{\boldsymbol{\beta}}_\lambda}$ and so $\text{MSE}_{\hat{\boldsymbol{\beta}}} > \text{MSE}_{\hat{\boldsymbol{\beta}}_\lambda}$ by Theorem 2.4.1. □

This result connects the two limit behaviours we have analysed in Equation (2.4) and Equation (2.6): by having a greater value of the tuning parameter, λ, the estimates are shrinked towards towards and during the process both the expected value and the variance are also pushed towards zero, but at different rates. This asymmetry leads the ridge to always reach a balance in the bias-variance trade-off, resulting in yielding a better performance than the OLS, in terms of a lower mean squared error. This also means that during fine-tuning procedures to find a value for the penalty parameter, λ, the value zero will always be rejected.

Following this results, we can conclude that when the aim is prediction, and hence it is indirectly required to reach the smallest mean squared error possible, the ridge estimator is always a better choice than the ordinary least squares one (when $\mathbf{X}^\top\mathbf{X}$ is of full rank), with no regard of the particular case of application. Moreover, this result reinforces the Gauss-Markov theorem: being the OLS the best linear unbiased estimator the only way to improve in terms of mean squared error is to move away from the class of the unbiased estimators, thus introducing bias.

As stated in Theorem 2.4.1, the result holds for all the formulations of the mean squared error that can be written as

$$\text{MSE}[\hat{\boldsymbol{\beta}}_\lambda] = \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})^\top\mathbf{A}(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})],$$

with the matrix $\mathbf{A} \succeq 0$. Hence it is also possible to demonstrate that the ridge estimator has the property to yield a lower value also when the mean squared error of the predicted values is used, when values of the tuning parameter, λ, greater than zero are selected.

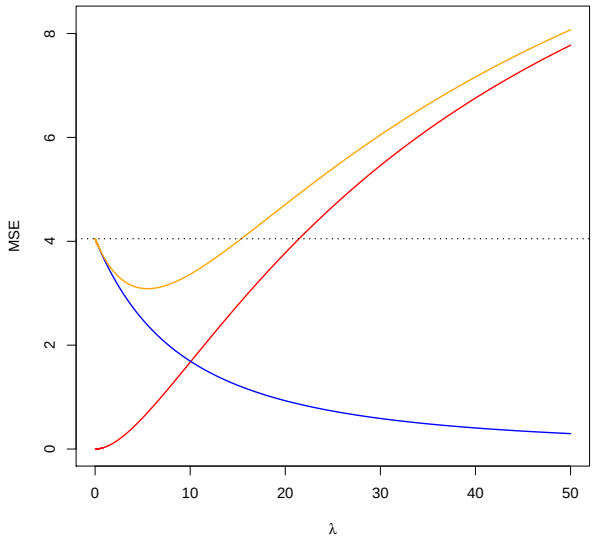


Figure 2.2: Bias-variance trade-off in ridge regression. The bias (in red), increases while the variance (in blue) decreases for greater valueso of the tuning parameter λ. As a result, the mean squared error always has a minimum point below the OLS mean squared error (dotted line).

For the remainder of this thesis, the attention will be turned on both the expected estimation error and the expected prediction error, which is defined as:

$$\begin{aligned}\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{E}[(\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda - \mathbf{X}\boldsymbol{\beta})], \\ &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})^\top\mathbf{X}^\top\mathbf{X}(\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta})].\end{aligned}$$

This type of mean squared error is equal to the expected estimation error (shown in equation 2.8) if and only if $\mathbf{X}^\top\mathbf{X} = \mathbf{I}_p$. The latter condition takes place when all the variables contained in $\mathbf{X}$ are perfectly independent from one another. Since this is not a very common condition, the two types of mean squared error will generally yield different values for λ, hinting that there might not be a single value of the tuning parameter λ which has to be generally preferred. Following the same steps shown for the expected estimation error, it is possible to demonstrate that the bias-variance decomposition of the expected prediction error is:

$$\begin{aligned}\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda] &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])^\top\mathbf{X}^\top\mathbf{X}(\hat{\boldsymbol{\beta}}_\lambda - \mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda])] + (\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta})^\top\mathbf{X}^\top\mathbf{X}(\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] - \boldsymbol{\beta}), \quad (2.10) \\ &= \sigma^2\text{tr}[\mathbf{X}\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top\mathbf{X}^\top] + \boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_p)^\top\mathbf{X}^\top\mathbf{X}(\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta}.\end{aligned}$$

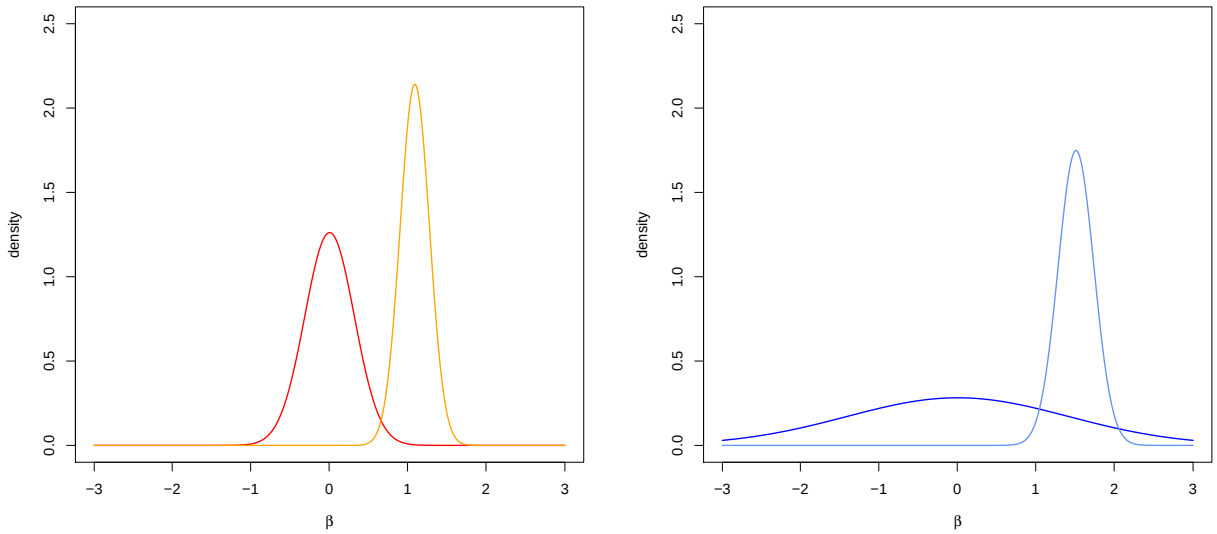
2.5 A Bayesian point of view

Ridge regression is also connected with Bayesian linear regression. In Bayesian regression, the parameters $\boldsymbol{\beta}$ and σ^2 of the linear model are considered to be random variables, keeping the design matrix $\mathbf{X}$ and the response vector $\boldsymbol{y}$ as non-stochastic elements. If we assume that the error term is distributed normally, the conjugate prior for the coefficients $\boldsymbol{\beta}$ is a multivariate normal:

$$\boldsymbol{\beta} \sim \mathcal{N}\left(\mathbf{0}_{p\times 1}, \frac{\sigma^2}{\lambda}\mathbf{I}_{p\times p}\right),$$

if the noise variance, σ^2, is assumed to be known. Alternatively, the conjugate prior for σ^2 would be an Inverse-Gamma distribution with hyperparameters not involving λ. In this framework, the penalty parameter λ becomes the precision parameter for the normal distribution: larger values of it will lead to a lower variance, hence the estimates will be pushed towards the mean, which is zero. This prior structure indicates that one of the main properties of ridge regression, the shrinkage behaviour of the estimates, can also translate into the Bayesian context.

In the Bayesian framework, the ridge estimate of $\boldsymbol{\beta}$ will be equivalent to the *maximum*



(a) (b)

Figure 2.3: Prior-posterior updating in ridge regression. On the left side a large value for λ is chosen ($\lambda = 10$), on the right side a small one ($\lambda = 0.5$). It is possible to notice how these values affect the precision of the prior and how such prior gets updated *caeteris paribus*.

a posteriori probability (MAP) of such vector. The posterior distribution of $\boldsymbol{\beta}$, using Bayes' Theorem is

$$\begin{aligned}\pi(\boldsymbol{\beta}|\boldsymbol{y},\mathbf{X}) &\propto \pi(\boldsymbol{\beta}) f(\boldsymbol{y},\mathbf{X}|\boldsymbol{\beta}), \\ &\propto \exp\left\{-\frac{\lambda}{2\sigma^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}\right\} \times \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})\right\}, \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\left[(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}\right]\right\}.\end{aligned} \tag{2.11}$$

Note that

$$\begin{aligned}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} &= \boldsymbol{y}^\top\boldsymbol{y} - \boldsymbol{\beta}^\top\mathbf{X}^\top\boldsymbol{y} - \boldsymbol{y}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{\beta}, \\ &= \boldsymbol{y}^\top\boldsymbol{y} + \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)\boldsymbol{\beta} \\ &\quad - \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top\boldsymbol{y} \\ &\quad - \boldsymbol{y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)\boldsymbol{\beta}, \\ &= \boldsymbol{y}^\top\boldsymbol{y} + [\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_\lambda]^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_\lambda] \\ &\quad - \hat{\boldsymbol{\beta}}_\lambda^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)\hat{\boldsymbol{\beta}}_\lambda.\end{aligned}$$

By removing the terms that do not depend from the parameters $\boldsymbol{\beta}$ and plugging this into 2.11

$$\pi(\boldsymbol{\beta}|\boldsymbol{y},\mathbf{X}) \propto \exp\left\{-\frac{1}{2\sigma^2}[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_\lambda]^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{p\times p})[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_\lambda]\right\},$$

which means that the posterior distribution of $\boldsymbol{\beta}$ is a multivariate normal distribution centered in $\hat{\boldsymbol{\beta}}_\lambda$. Being the posterior another normal distribution, the MAP, which is the mode by definition, it is also the posterior mean.

The tuning parameter, λ, still affects the precision of the posterior, its variance being $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}$. Larger values of this parameter would then give posterior estimates that show more shrinking towards zero while having less variance.

2.6 Fine-tuning of the tuning parameter

Starting from a mean squared error formulation, there will always be a value of the tuning parameter, λ, that minimizes it, but it will inevitably depend on the unknown quantities σ^2 and $\boldsymbol{\beta}$. For this reason, that particular value of the tuning parameter is referred to as the *oracle value*. Estimating the oracle value is then crucial in order to build a regression estimator that is set to yield to the lowest mean squared error as possible. To accomplish

this task, several different methods can be used. Stemming from the Bayesian view of ridge regression, one possible technique is maximum marginal likelihood estimation. The widely known technique of cross-validation could also be used to find a value for the tuning parameter, but its aim it is not represented by the estimation of the oracle value. Eventually, the oracle value of the tuning parameter, λ, can be estimated through the minimization of the theoretical expression of the expected error of the ridge estimator, with the aid of suitable plug-in estimates of the unknown quantities involved.

In the rest of the thesis this last method would be at the centre of the analysis for its link to other formulation of the ridge method and its use as a risk function to minimize when introducing model selection into ridge, as we will show in Chapter 3. We will also compare this method to cross-validation, as it is the most common technique used to fine tune the tuning parameter of the ridge estimator, even if its aim is not to estimate the oracle value but to find the value of the tuning parameter that yields to the best predictions, given the data at disposal.

2.6.1 Maximum marginal likelihood

In a Bayesian framework it is possible to reach an estimate of λ by maximising its marginal likelihood. The prior distribution of $\boldsymbol{\beta}$ ought to be a normal random variable with these parameters:

$$\boldsymbol{\beta} \sim \mathcal{N}\left(\mathbf{0}_{p\times 1}, \frac{\sigma^2}{\lambda}\mathbf{I}_{p\times p}\right).$$

Following the simple reparametrization $\lambda = \frac{\sigma^2}{\tau^2}$, an estimate of τ might be achieved, assuming that σ^2 is known or very well estimated. Since also $\boldsymbol{y}|\boldsymbol{\beta}$ is also normally distributed, with mean $\mathbf{X}\boldsymbol{\beta}$ and covariance matrix $\sigma^2\mathbf{I}_{n\times n}$, the marginal likelihood of τ, can be obtained by first computing the likelihood funciton of $\boldsymbol{y}$ (which depends also on τ) and then maximizing it. The former task is accomplished by solving the integral

$$f(\boldsymbol{y}) = \int f(\boldsymbol{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta})d\boldsymbol{\beta},$$

which leads to

$$\boldsymbol{y} \sim \mathcal{N}\left(\mathbf{0}_{n\times n}, \sigma^2\mathbf{I}_{n\times n} + \tau^2\mathbf{X}\mathbf{X}^\top\right).$$

Hence the maximum marginal estimator of τ from which then derive the estimate of λ is the

value that maximizes the likelihood, or log-likelihood, of $\boldsymbol{y}$:

$$\begin{aligned}\hat{\tau} &= \arg\max_{\tau}\{\log[f(\boldsymbol{y},\tau)]\},\\ &= \arg\max_{\tau}\left\{\log\left[|\mathbf{I}_n+\tau^2\mathbf{X}\mathbf{X}^\top|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\boldsymbol{y}^\top(\mathbf{I}_n+\tau^2\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y}\right)\right]\right\},\\ &= \arg\max_{\tau}\{-\log|\mathbf{I}_n+\tau^2\mathbf{X}\mathbf{X}^\top|-\boldsymbol{y}^\top(\mathbf{I}_n+\tau^2\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y}\}.\end{aligned}$$

It is not possible to reach a solution for this expression analytically, hence the use of a numerical maximizer is needed. However, this does not constitute a problem for the use of such procedure since a result can easily computed following this way.

2.6.2 Cross-validation

When the aim is to assess the prediction performances of a model, the best possible way to accomplish the task is to fit the model on present data and then predict new outcomes, as soon as it is possible to have access on a new, independent, sample. In some occasions, new data could not be accessed to easily, hence cross-validation can be used to solve the problem. The cross-validation technique, in fact, tries to solve the issue by constructing several sets of hold-out data, simulating independent new samples. Once an hold-out sample, or test set, is created, a model is fit on the rest, the training set, and then evaluated in terms of its prediction on the former. Since cross-validation permits to estimate the prediction performances of a method, it also can be used to perform fine-tuning of parameters, if it is needed. In fact, several values for the tuning parameters can be tested through cross-validation, then selecting the ones that yield to the lowest estimate of the prediction error.

Cross-validation involves the use of several hold-out samples because the procedure might be unstable when using few splits. In these situations, the prediction performances would be dependent on the actual splits used. Also using several random hold-out samples might be problematic: while splitting the data randomly there is no control on which data points will fall in the training or test set at each iteration, this way some influential observations could be never used to fit the model or be tested on. Cross-validation, or more precisely K-fold-cross-validation, represents a solution to this issue.

When this technique is used, data are split beforehand in in K parts, or folds, then all of them are used in the role of the test set only once. At each iteration, the training set will be formed by the union of the remaining folds. Eventually the prediction performances are estimated averaging the results collected onto the K training sets used. The K-fold-cross-

validation algorithm in regards to ridge regression starts with setting up a grid of L values for the tuning parameters λ, and then:

1. Data is divided into K folds

2. A ridge regression model is then estimated for every λ value of the grid, removing one fold at a time to form the training set associated to that fold (here I indicate with $\mathbf{X}_{-k}$ the design matrix where k-th fold is removed):

$$\hat{\boldsymbol{\beta}}_{\lambda_l,k} = (\mathbf{X}_{-k}^\top \mathbf{X}_{-k} + \lambda_l \mathbf{I}_p)^{-1} \mathbf{X}_{-k}^\top \boldsymbol{y}_{-k} \quad \text{for } l = 1, \dots, L \text{ and } k = 1, \dots, K$$

3. For each λ average the prediction error over the K folds and then take the value that yields to the least average prediction error.

As stated by James et al. [2013], the number of folds used, K, covers a crucial role in regards to the estimation of prediction error. When more folds are used, every training set used in the cross-validation procedure would contain a similar amount of information as the whole dataset. For this reason, the estimated prediction error will have low bias but, on the other hand, will have high variance, since the predictions produced at each step will be highly correlated with each other. The bias-variance trade-off also applies in the framework of prediction error estimation through cross-validation. The trade-off become apparent when the two limit case of cross-validation are analysed. When the minimum number of folds is taken, which is two, the two estimates of the prediction error will be independent and hence their average would have no variance. By contrast, when every fold used is formed by a single unit, as in the case of *leave-one-out* cross-validation, the average prediction error will be constituted by several positively correlated estimates. In this case, the average of these estimates will yield to an estimate of prediction error which is characterized by low bias but high variance. For this reason, leave-one-out cross-validation is to be preferred when unbiased estimates of the prediction error are needed, even though in common practice 5-fold and 10-fold cross-validation are the most widely used settings.

As a concluding remark, the main feature of cross-validation is its flexibility: by not requiring any theoretical derivation for the tuning parameters of a model or its mean squared error (or other metrics), it is always possible to apply the algorithm testing several values for the tuning parameters and then assessing their performances.

2.6.3 Minimization of the mean squared error

Starting from the expression of the expected estimation error in Equation (2.9) and of the expected prediction error in Equation (2.10), it possible to define the oracle tuning of the penalty parameter, λ:

Definition 2.6.1. *Given a mean squared error formulation, the oracle tuning parameter associated with it is its minimand*

$$\lambda_{\text{MSE}} = \arg\min_{\lambda}\{\text{MSE}_{\lambda,\boldsymbol{\beta},\sigma^2}\},$$

when the parameters $\boldsymbol{\beta}$ and σ^2 are known.

Even if the parameters $\boldsymbol{\beta}$ and σ^2 are not known generally, the oracle value can still be estimated, by using plug-in (or pilot) estimates for the unknown quantities $\tilde{\sigma}^2$ and $\tilde{\boldsymbol{\beta}}$. When the dimensionality is low, whenever $p < n$, a natural choice for the plugins are the ordinary least squares estimates. In this framework, the estimates of the regression coefficients contained by $\boldsymbol{\beta}$ are given by Equation (2.1), whereas the variance of the error term, σ^2, is estimated by:

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{p}(y_i - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}})^2.$$

When it is not possible, nor useful, to use the ordinary least squares estimates, pilot estimates might be also derived from the ridge method. In that case, the estimator for $\boldsymbol{\beta}$ will assume the form described in Equation (2.2), with the value of the tuning parameter, λ, chosen by cross-validation ($\hat{\lambda}_{cv}$), yielding to this variance estimate:

$$\hat{\sigma}^2_{\hat{\lambda}_{cv}} = \frac{1}{n - df(\hat{\lambda}_{cv})}\sum_{i=1}^{p}(y_i - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}_{\hat{\lambda}_{cv}})^2,$$

where the *effective* degrees of freedom of $\hat{\lambda}_{cv}$ are computed as $df(\hat{\lambda}_{cv}) = \text{tr}[\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^\top]$.

Hence we can define the estimates of the oracle value of the tuning parameter:

Definition 2.6.2. *The estimate of the oracle tuning parameter of a given mean squared error formulation, is obtained by minimizing that expression, substituting the unknown parameters with their pilot estimates $\tilde{\boldsymbol{\beta}}$ and $\tilde{\sigma}^2$*

$$\hat{\lambda}_{\widehat{\text{MSE}}} = \arg\min_{\lambda}\{\widehat{\text{MSE}}_{\lambda,\tilde{\boldsymbol{\beta}},\tilde{\sigma}^2}\}.$$

Eventually, the ridge estimator that can be obtained by this procedure involves the use of this estimated value for the tuning parameter:

$$\hat{\boldsymbol{\beta}}_{\lambda} = (\mathbf{X}^{\top}\mathbf{X} + \hat{\lambda}_{\widehat{\mathrm{MSE}}}\,\mathbf{I}_n)^{-1}\mathbf{X}^{\top}\boldsymbol{y}.$$

As a concluding remark, it is possible to choose other estimates as plugins such as lasso estimates, or principal component regression estimates [Hellton and Hjort, 2018] but when the situation suggests to use ridge regression, it is understandable to use the same estimator as the plug-in, or the OLS, since the former is an improvement of the latter, as stated in the beginning of this chapter.

2.7 Simulation study

Once a value for the tuning parameter λ is chosen, the ridge estimator can be finally computed. Since the point of using the MSE minimization technique is to yield the lowest *expected* error, and since such quantities depend on a correct estimation of the oracle tuning parameter λ, we conducted a simulation study to assess this property. The two main questions that should be answered are whether the technique effectively estimates the oracle value of the tuning parameter λ and whether or not the ability of minimizing the expected mean squared error could translate in being competitive prediction-wise with its main competitor, cross-validation. In order to accomplish such tasks, synthetic data were used throughout the simulation process.

Using the formulations given earlier in this chapter inevitably brings to several computational inefficiencies. Before presenting the results, are first shown some practical considerations on how to reduce the computations necessary. This permitted to sensibly shorten the computational time requested.

2.7.1 Cross-validation shortcut

One particular version of the cross-validation procedure is the so called *leave one out cross-validation* (abbreviated LOOCV). This type of cross-validation is obtained when the number of folds is equal to the number of the observations n, hence prediction error is estimated by leaving one unit out of the dataset at each step. Using the maximum number possible of folds, this procedure is normally computationally heavy. In ridge regression, the leave-one-out prediction error can be computed without performing the algorithm, so leave-one-out cross-validation can be reduced to a simpler form.

In ridge regression, the predicted values for a fixed λ can be written as

$$\hat{\boldsymbol{y}}_\lambda = \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_n)^{-1}\mathbf{X}^\top\boldsymbol{y} = \mathbf{H}_\lambda\boldsymbol{y},$$

with $\mathbf{H}_\lambda = \mathbf{X}\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$.

At each step, after removing the ith unit from the dataset, the data matrix will be denoted $\mathbf{X}_{-i}$ and the response variable vector $\boldsymbol{y}_{-i}$. Then the estimated regression model for each leave-one-out cross-validation step is

$$\hat{\boldsymbol{\beta}}_{-i,\lambda} = (\mathbf{X}_{-i}^\top\mathbf{X}_{-i} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}_{-i}^\top\boldsymbol{y}_{-i}.$$

The error of each removed unit is then

$$\boldsymbol{e}_{-i} = \boldsymbol{y}_{-i} - \boldsymbol{x}_{-i}^\top\hat{\boldsymbol{\beta}}_{-i,\lambda}.$$

By the Sherman-Morrison-Woodbary formula [Petersen et al., 2008] for matrix inverses, one has the relation

$$(\mathbf{X}_{-i}^\top\mathbf{X}_{-i} + \lambda\mathbf{I}_p)^{-1} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1} + \frac{(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}}{1 - \mathbf{H}_{ii,\lambda}},$$

where the diagonal of the $\mathbf{H}$ matrix is denoted $\mathbf{H}_{ii,\lambda} = \boldsymbol{x}_i^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i$. Together with $\mathbf{X}_{-i}^\top\boldsymbol{y}_{-i} = \mathbf{X}^\top\boldsymbol{y} - \boldsymbol{x}_i\boldsymbol{y}_i$, the regression coefficients of each estimated cross-validation model is given

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{-i,\lambda} &= \left[(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1} + \frac{(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}}{1 - \mathbf{H}_{ii}}\right](\mathbf{X}^\top\boldsymbol{y} - \boldsymbol{x}_i\boldsymbol{y}_i), \\
&= \hat{\boldsymbol{\beta}}_\lambda - \left[\frac{(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i}{1 - \mathbf{H}_{ii}}\right]\left[\boldsymbol{y}_i(1 - \mathbf{H}_{ii}) - \boldsymbol{x}_i^\top\hat{\boldsymbol{\beta}}_\lambda + \mathbf{H}_{ii}\boldsymbol{y}_i\right], \\
&= \hat{\boldsymbol{\beta}}_\lambda - (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i\boldsymbol{e}_i/(1 - \mathbf{H}_{ii}),
\end{aligned}$$

with the following expression for each cross-validated error

$$\begin{aligned}
\boldsymbol{e}_{-i} &= \boldsymbol{y}_i - \boldsymbol{x}_i^\top\hat{\boldsymbol{\beta}}_{-i,\lambda} = \boldsymbol{y}_i - \boldsymbol{x}_i^\top\left[\hat{\boldsymbol{\beta}}_\lambda - (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\boldsymbol{x}_i\boldsymbol{e}_i/(1 - \mathbf{H}_{ii})\right], \\
&= \boldsymbol{e}_i + \mathbf{H}_{ii}\boldsymbol{e}_i/(1 - \mathbf{H}_{ii}) = \boldsymbol{e}_i/(1 - \mathbf{H}_{ii}).
\end{aligned}$$

The mean cross-validation criterion can then be derived as

$$\mathrm{CV}_\lambda = \sum_{i=1}^{n}\boldsymbol{e}_{-i}^2 = \sum_{i=1}^{n}\left(\frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i}{1 - \mathbf{H}_{ii,\lambda}}\right)^2,$$

where $\mathbf{H}_{ii,\lambda}$ are the diagonal elements of the $\mathbf{H}$ matrix. Again, the minimand of $cv(\lambda)$ as a function of λ gives the cross-validation tuning parameter

$$\hat{\lambda}_{cv} = \arg\min_{\lambda} \mathrm{CV}_{\lambda} = \arg\min_{\lambda} \sum_{i=1}^{n} \left(\frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i}{1 - \mathbf{H}_{ii,\lambda}} \right)^2 .$$

Leave-one-out cross-validation will be used later in this work for both its property to produce unbiased estimates of the prediction error and this formulation, which shortens the computational time requested by a significant amount.

2.7.2 Singular value decomposition of the mean squared error

Also the minimization of the mean squared error can be a computationally intensive task, as it involves several full matrix inversions. There is a way to avoid any full inversion, using the singular value decomposition of the design matrix $\mathbf{X}$. Using this decomposition, $\mathbf{X}$ can be viewed as the matrix product $\mathbf{UDV}^\top$ where $\mathbf{U}$ is a $n \times p$ orthogonal matrix, $\mathbf{V}$ is a $p \times p$ orthogonal matrix and $\mathbf{D}$ is a diagonal $p \times p$ matrix containing on its diagonal the p singular values of the design matrix, noted as $d_1, \ldots, d_p$. Eventually, by definition, we have that $\mathbf{U}^\top\mathbf{U}$, $\mathbf{V}^\top\mathbf{V}$ and $\mathbf{V}\mathbf{V}^\top$ are equal to $\mathbf{I}_p$.

Plugging in the pilot estimates $\tilde{\sigma}^2$ and $\tilde{\boldsymbol{\beta}}$ into the MSE expression as seen in Equation (2.9) and writing the design matrix $\mathbf{X}$ in its singular value decomposition, we obtain

$$\begin{aligned}
\mathrm{MSE}[\hat{\boldsymbol{\beta}}_\lambda] &= \tilde{\sigma}^2 \mathrm{tr}[\mathbf{W}_\lambda (\mathbf{X}^\top\mathbf{X})^{-1} \mathbf{W}_\lambda^\top] + \tilde{\boldsymbol{\beta}}^\top (\mathbf{W}_\lambda - \mathbf{I}_p)^\top (\mathbf{W}_\lambda - \mathbf{I}_p) \tilde{\boldsymbol{\beta}}, \\
&= \tilde{\sigma}^2 \mathrm{tr}[\mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top \mathbf{V}\mathbf{D}^{-2}\mathbf{V}^\top \mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top] \\
&\quad + \tilde{\boldsymbol{\beta}}^\top [\mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top - \mathbf{V}\mathbf{V}^\top]^\top [\mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top - \mathbf{V}\mathbf{V}^\top] \tilde{\boldsymbol{\beta}}, \\
&= \tilde{\sigma}^2 \mathrm{tr}[\mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-2}\mathbf{V}^\top] + \tilde{\boldsymbol{\beta}}^\top \mathbf{V} [\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1} - \mathbf{I}_p]^2 \mathbf{V}^\top \tilde{\boldsymbol{\beta}}, \\
&= \tilde{\sigma}^2 \mathrm{tr}[\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-2}] + \tilde{\boldsymbol{\beta}}^\top \mathbf{V} [\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I}_p)^{-1} - \mathbf{I}_p]^2 \mathbf{V}^\top \tilde{\boldsymbol{\beta}}.
\end{aligned} \tag{2.12}$$

At this stage every full matrix inversion previously required has disappeared from the formulation of the mean squared error. The only matrix inversions required are now inversions of diagonal matrices which are less demanding tasks, as they can be computed by substituting the elements present on the diagonal with their reciprocals. In particular, when $\mathbf{X}$ is of full rank, its singular values will be all non-zero, otherwise the tuning parameter λ will be asked to counterbalance the presence of null singular values.

It is possible to show that this expression can be further simplified as a sum of sums. The two terms that will be in this further manipulation are the trace of a matrix, which

is a sum by definition, and the multiplication of a vector by a diagonal matrix and then by itself. The latter term consists then in the sums of the multiplications of the squared elements of the vector by the corresponding elements of the diagonal matrix. Note that the two aforementioned diagonal matrices have as ith term on the diagonal

$$\{\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-2}\}_{ii} = \left(\frac{d_i}{d_i^2+\lambda}\right)^2 \quad \text{for } i=1,\ldots,p,$$
$$\{\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p\}^2_{ii} = \lambda^2\left(\frac{1}{d_i^2+\lambda}\right)^2 \quad \text{for } i=1,\ldots,p,$$

and by defining the $p\times 1$ vector $\boldsymbol{\nu}=\mathbf{V}^\top\tilde{\boldsymbol{\beta}}$, the mean squared error expression can be finally written as

$$\text{MSE}[\hat{\boldsymbol{\beta}}_\lambda] = \tilde{\sigma}^2\sum_{i=1}^{p}\left(\frac{d_i}{d_i^2+\lambda}\right)^2 + \lambda^2\sum_{i=1}^{p}\left(\frac{\nu_i}{d_i^2+\lambda}\right)^2.$$

Applying the same decomposition to the mean squared prediction error presented in Equation (2.10), we obtain

$$\begin{aligned}
\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda] =& \tilde{\sigma}^2\text{tr}[\mathbf{X}\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top\mathbf{X}^\top] + \tilde{\boldsymbol{\beta}}^\top(\mathbf{W}_\lambda-\mathbf{I}_p)^\top\mathbf{X}^\top\mathbf{X}(\mathbf{W}_\lambda-\mathbf{I}_p),\tilde{\boldsymbol{\beta}} \qquad (2.13)\\
=& \tilde{\sigma}^2\text{tr}[\mathbf{V}\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top\mathbf{V}\mathbf{D}^2\mathbf{V}^\top\mathbf{V}\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}\mathbf{V}^\top\mathbf{V}\mathbf{D}^{-2}\mathbf{V}^\top]\\
&+\tilde{\boldsymbol{\beta}}^\top\mathbf{V}[\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p]^\top\mathbf{V}^\top\mathbf{V}\mathbf{D}^2\mathbf{V}^\top\mathbf{V}[\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p]\mathbf{V}^\top\tilde{\boldsymbol{\beta}},\\
=& \tilde{\sigma}^2\text{tr}[\mathbf{V}\mathbf{D}^4(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-2}\mathbf{V}^\top] + \tilde{\boldsymbol{\beta}}^\top\mathbf{V}\mathbf{D}^2[\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p]^2\mathbf{V}^\top\tilde{\boldsymbol{\beta}},\\
=& \tilde{\sigma}^2\text{tr}[\mathbf{D}^4(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-2}] + \tilde{\boldsymbol{\beta}}^\top\mathbf{V}\mathbf{D}^2[\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p]^2\mathbf{V}^\top\tilde{\boldsymbol{\beta}}.
\end{aligned}$$

With this formulation of the mean squared error, we obtain an expression that can also be simplified in the same fashion as before. The two diagonal matrices involved now have as ith diagonal element:

$$\{\mathbf{D}^4(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-2}\}_{ii} = \left(\frac{d_i^2}{d_i^2+\lambda}\right)^2 \quad \text{for } i=1,\ldots,p,$$
$$\{\mathbf{D}^2[\mathbf{D}^2(\mathbf{D}^2+\lambda\mathbf{I}_p)^{-1}-\mathbf{I}_p]^2\}_{ii} = \lambda^2\left(\frac{d_i}{d_i^2+\lambda}\right)^2 \quad \text{for } i=1,\ldots,p.$$

Hence, by using the same $\boldsymbol{\nu}$ vector as before, the simplified version of the expected prediction error becomes:

$$\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda] = \tilde{\sigma}^2\sum_{i=1}^{p}\left(\frac{d_i^2}{d_i^2+\lambda}\right)^2 + \lambda^2\sum_{i=1}^{p}\left(\frac{\nu_i d_i}{d_i^2+\lambda}\right)^2. \qquad (2.14)$$

2.7.3 Results: minimization of expected mean squared error

In order to simulate the expected mean squared error, the simulation process starts with the creation of a fixed design matrix $\mathbf{X}$ and the setting of a coefficients vector $\boldsymbol{\beta}$. After that, 2000 response vectors $\boldsymbol{y}$ are simulated following the linear function $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with the error term drawn from a normal distribution with mean 0 and variance σ^2 equal to 1. In the simulation study we assessed the behaviour of both the expected estimation error and the expected prediction error, depending also from which plug-in estimates were used.

In the first simulation study, the main goal is to assess which plug-in should be preferred to estimate the oracle value of λ under different situations. We identified four frameworks that differ in regards to the signal strength of the regression coefficients present in $\boldsymbol{\beta}$ and the correlation matrix $\boldsymbol{\Sigma}$ of the fixed design matrix $\mathbf{X}$. Hence, the design matrix is drawn from $\mathcal{N}(0, \sigma^2\boldsymbol{\Sigma})$:

1. $\boldsymbol{\beta} = (0.5, -0.5, 0.1, -0.1, \dots)$ and $\mathbf{X}$ with $\boldsymbol{\Sigma}_{ii} = 1$ for $i = 1, \dots, n$ and $\boldsymbol{\Sigma}_{ij} = 0.2$ for $i \neq j$ and dimensions $n = 100$ and $p = 40$,

2. $\boldsymbol{\beta} = (3, -3, 1, -1, \dots)$ and $\mathbf{X}$ with same covariance structure as in Case 1,

3. $\boldsymbol{\beta} = (0.5, -0.5, 0.1, -0.1, \dots)$ and $\mathbf{X}$ with with $\boldsymbol{\Sigma}_{ii} = 1$ for $i = 1, \dots, n$ and $\boldsymbol{\Sigma}_{ij} = 0.9$ for $i \neq j$ of dimensions $n = 50$ and $p = 40$,

4. $\boldsymbol{\beta} = (3, -3, 1, -1, \dots)$ and $\mathbf{X}$ with same covariance structure as in Case 3.

Barring the high dimensional case, reached when $n < p$, in which it is impossible to estimate the $\boldsymbol{\beta}$ vector through the ordinary least squares method, one would expect that the OLS plug-in is set to perform better when the signal strength of true $\boldsymbol{\beta}$ vector is large since it is introduces no shrinking. Since the ordinary least squares estimator could be unstable when $\mathbf{X}$ suffers from high collinearity, it is expected to be a better choice when the covariance matrix of the data matrix is similar to the identity matrix. Conversely, the ridge estimator should be preferred as a plug-in when the size of the coefficients is smaller, and when the covariance structure of the data matrix shows high correlation between the covariates.

The results are reported in Table 2.1. The values in the table refers to the average mean squared errors scored through the 2000 iterations on the criterion of choice. This way, it is possible to assess how the plug-in estimates work in minimizing the criterion chosen, compared to the oracle tuning, which is expected to yield the lowest error in all cases. In all cases both plug-ins yield errors that are very similar to the oracle tuning ones, in Case 1

and Case 2 these values are virtually the same. In Case 3 and in Case 4, ridge estimates are expected to be better as plug-ins, given the dimensionality of the setting and the correlation structure, but we found that if the signal strength is large enough, also the ordinary least squares estimates are viable plug-in estimates.

For this reason, in Figure 2.4 are presented the density functions of the estimated tuning parameters, computed using the values produced through the 2000 iterations. These density functions refer to the transformation of the tuning parameter $\frac{\lambda}{1+\lambda}$, this way the value that those parameters can assume ranges from 0 to 1 and hence all methods become easily comparable in their choices. The most interesting results are the ones obtained in the setting with a lower signal strength, such as in Case 1 and Case 3. Turning the attention on Figure 2.4a, it is possible to notice how the ridge plug-in could be a better choice to estimate the oracle value of the expected estimation error. In this particular setting, its property to shrink the estimates could help this estimator in capturing better the signal in $\boldsymbol{\beta}$. When the correlation is high, the results shown in Figure 2.4c report that the OLS plug-in is clearly the wrong choice, as expected, while the ridge plug-in tends to overestimate the oracle value. When the signal strength increases, the presence of high correlation in the data does not impact the estimates of the oracle value in the same way, as it can be deduced from Figure 2.4c and Figure 2.4d. The latter, however, shows that in this case the expected prediction error criterion consistently underestimates the oracle value of the tuning parameter, λ, as all of the density functions show a positive skewness. The larger signal strength might be responsible for inducing the criterion to shrink less than necessary.

	Est-Or	Est-OLS	Est-R	Pred-Or	Pred-OLS	Pred-R
Case 1	2.2046*	2.2169*	2.2173*	0.3923	0.3928	0.3928
Case 2	2.2583*	2.2584*	2.2584*	0.4015	0.4015	0.4015
Case 3	0.0336	0.0742	0.0414	0.5801	0.6318	0.6253
Case 4	0.1031	0.1204	0.1203	0.7749	0.7827	0.7833

Table 2.1: Simulated criteria values. The wording "Est" and "Pred" indicates which criterion was minimized, with the first indicating the expected estimation error and the second the expected prediction error. The wording after the dash indicates which plug-in estimates for $\boldsymbol{\beta}$ and σ^2 was used. "Or" indicates the true value of the unknown quantities, leading to the oracle, "OLS" the ordinary least squares and "R" the ridge ones. Numbers indicated with * are meant to be multiplied by 10^{-3}.

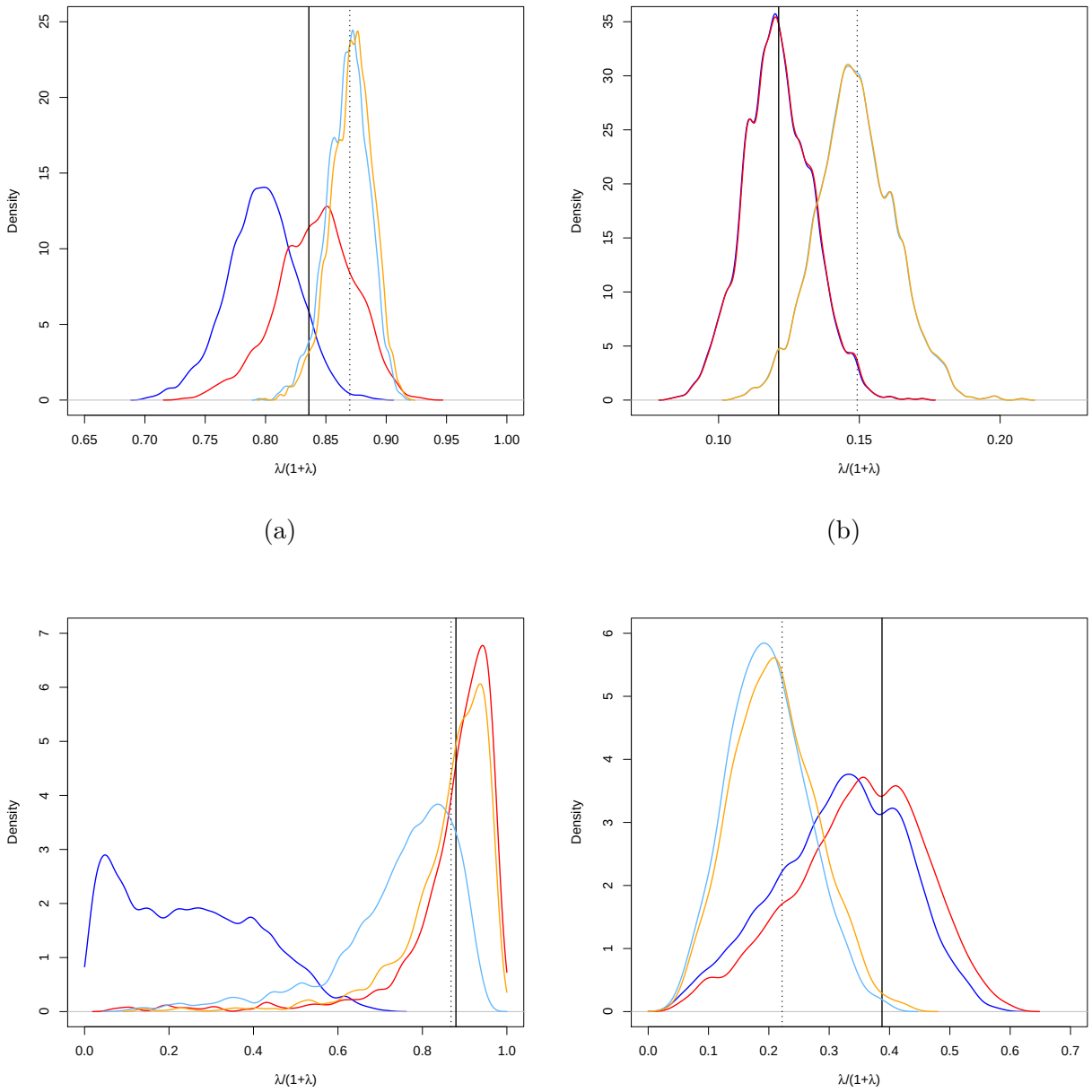


Figure 2.4: Estimated values of the tuning parameter, λ. The parameter is transformed in $\frac{\lambda}{1+\lambda}$ in order to obtain comparable values among all methods as the transformed value range from 0 to 1. In All plots, blue (OLS plug-in) and red (ridge plug-in) lines indicate the results when the criterion used is the expected estimation error, light blue (OLS plug-in) and orange (ridge plug-in) refer to the expected prediction error. Solid line indicates the oracle value of the former criterion, the dashed one refers to the latter. In (a) the data matrix used and the beta vector are the same as in Case 1, in (b) as Case 2, in (c) as Case 3 and in (d) as in Case 4.

	LOOCV	Est-Or	Est-OLS	Est-R	Pred-Or	Pred-OLS	Pred-R
Case 1	0.6724	0.667	0.6732	0.6742	0.6667	0.6715	0.6719
Case 2	0.6798	0.679	0.6792	0.6792	0.679	0.6791	0.6791
Case 3	2.4988	2.1414	3.0009	2.5889	2.1728	2.5993	2.4517
Case 4	4.4594	4.1754	4.4613	4.4961	4.2161	4.409	4.4167

Table 2.2: Predictive perfomance comparison. The wording "LOOCV" refers to the leave-one-out cross-validation procedure. The criteria and plug-in used to minimize the mean squared error expressions of ridge regression are signaled as in Table 2.1.

2.7.4 Results: comparison of prediction error with LOOCV

The second goal of the simulation study was to assess whether or not these methods could also be competitive with the cross-validation technique in yielding a low prediction error. The main settings used are the same as before. In every iteration of the simulation, first was generated a response variable vector $\boldsymbol{y}$, then the data matrix was randomly split in a training and a test set with equal size.

Every version of the ridge estimator we wanted to test is then computed on the training set and used to perform prediction on the test set. The prediction error is computed by

$$\text{MSE}[\mathbf{X}_{test}\hat{\boldsymbol{\beta}}_\lambda] = \frac{1}{m}\sum_{j=1}^{m}\left[(\boldsymbol{x}_{test,\,j}^\top\boldsymbol{\beta} - \boldsymbol{x}_{test,\,j}^\top\hat{\boldsymbol{\beta}}_\lambda)^2\right] \quad \text{for} \quad j = 1, \dots, m,$$

where $\boldsymbol{\beta}$ is the true regression parameter and m is the sample size of the test test. In this setting, then, we are interested to assess the out-of-sample prediction error yielded by the techniques we presented earlier. To do that, at every iteration the fixed design matrix is randomly split in a training set and in a test set having the same size.

The results are far enough apart to declare a clear winner, even though they contain some surprises. First of all, the oracle values of the tuning parameter perform consistently better than any other method in every situation tested. In addition, all of the methods tested are sensitive to the signal strength, but our methods seem to respond better, as in some cases they register a larger improvement than cross-validation when the signal strength increases. In all Cases other than the first, oracle tuning of the expected estimation error yields a lower prediction error than the oracle tuning of the expected prediction error, which is strange. In order to reverse this apparent contrast, when the expected prediction error is minimized through plug-in estimates, it is possible to obtain lower errors than the one

obtained by minimizing the expected estimation error. Hence, the expected prediction error has to be preferred in this setting, as the oracle tuning of the expected estimation error remains unknown in any practical application.

Estimating the oracle value of the tuning parameter seems to be an alternative to cross-validation which is worth exploring. Judging from the prediction error scored in several situations, this method could be competitive with cross-validation.

Chapter 3

Adding model selection to ridge regression

When the data possesses multiple covariates, it is possible to build several models, each one of them characterized by the use of a certain number of covariates. These models might be also called submodels because they are necessarily built on a subset, S, of the p covariates contained in the full design matrix $\mathbf{X}$. The main goal of model selection is to identify which subset of covariates is contains the variables that are linked to the outcome, hence finding the true model underlying the data generating process. This also means that every model selection method starts with the important assumption by which the true model is effectively present in the list of submodels which the method can choose from. Such assumption might be also relaxed, one could ask a model selection method to find a model which is sufficiently close to the true model by a metric, such as the Kullback-Leibler distance, as reported in Claeskens and Hjort [2008]. This task may cover an important role when the amount of covariates is large, or when it is likely that the data has some *noise variables* hidden in it. These covariates are may in fact constitute a problem because they have no real connection with the outcome, hence they do not belong in the true model.

In this chapter we will present an overview upon some of the most used model selection techniques, and eventually we will show how to implement the FIC methodology into ridge regression, hence creating a ridge regression method that it is also capable to perform covariate selection.

3.1 An overview of some model selection methods

Starting with a design matrix with p covariates, one may ask if all of them must be included in every model. Starting from p covariates, it is possible to build 2^p different models, each one of them referred to a specific subset S of the p covariates we start with. A first reason behind the choice to perform model selection is certainly the will to identify a parsimonious model that can offer a simpler interpretation of the response variable. With datasets that offer a very wide range of covariates another reason could be avoiding noise. In some situations, in fact, there may be the presence of covariates that appear to carry useful information to explain the output, but such connection is a product of chance chance. It is hence preferable to exclude them from the analysis. In both of these situations, any model selection method has the need to rank the models which is fed with, in order to be able to select a preferred one.

A first way to select models is to use an information criterion, the most famous of them being Akaike's Information Criterion, which is also referred to as AIC, introduced by Akaike [1973] and the Bayesian Information Criterion, which is referred to as BIC, introduced by Schwarz et al. [1978] and Akaike [1977], Akaike [1978]. Such criteria assign a score to each models by penalizing the value of the likelihood with a quantity proportional to the size of the model in term of number of parameters estimated by those models. The best model will then be the model that has the highest AIC or BIC score.

It is also possible to use a penalized regression technique to perform model selection, which is the lasso [Tibshirani, 1996]. As we will see, by using the L1 norm of the coefficients as penalization, the model will operate both shrinkage and variable selection at the same time because, as a result of its particular penalization, some coefficients can be estimated to be exactly zero, hence removing the associated covariates them from the final model.

Finally, another information criterion that could be used is the one introduced by Claeskens and Hjort [2008], which is the focussed information criterion, also termed FIC. Such criterion stems from the idea that a model can also be selected not only by its overall fit, but with respect of its performance on a particular quantity of interest, namely the *focus*, which will be evaluated in terms of estimation error.

3.1.1 AIC and BIC criteria

Starting from a parametric model $\mathcal{M}$, and indicating the number of parameters estimated by that model with $\dim(\mathcal{M})$, Akaike's information criterion is defined as:

$$\text{AIC}(\mathcal{M}) = 2\text{log-likelihood}_{max}(\mathcal{M}) - 2\dim(\mathcal{M}).$$

When this criterion is used to select a model, the one model that is deemed to be chosen is the one that has the highest AIC score. The addition of a penalty proportional to the number of estimated parameters to the likelihood serves as a counterbalance of the latter quantity. As the dimension of the model increases, a criterion involving only the likelihood would always select the biggest model possible as the likelihood can be inflated by simply estimating more parameters. With this type of penalty, a new parameter is added to the model only if the new likelihood increases by more than a certain amount, which in this case is one.

The Bayesian Information Criterion has the same shape as the AIC but is characterized by the use of another penalization term, which now depends also from the sample size n:

$$\text{BIC}(\mathcal{M}) = 2\text{log-likelihood}_{max}(\mathcal{M}) - (\log n)\dim(\mathcal{M}).$$

This different penalization makes the BIC a more parsimonious criterion because the additions of new parameters have now to induce a larger increase in the likelihood of the model. The main difference between the AIC and BIC involves their behaviour when the sample size, n, increases. If the true model the method is set to find is supposed to be actually present is the group of submodels scored with AIC or BIC, the latter criterion will select this true model almost surely, for diverging values of n. It is also possible to extend this result even when the true model is not on the list of submodels which the method can choose from, but there is a sufficiently near model, according to the Kullback-Leibler distance. Such property of the BIC criterion is called *consistency*. When the sample size is small, however, the strong penalty to the model complexity imposed by the BIC will tend to select models that might result be too parsimonious. In these situations, it is preferable to use Akaike's information criterion.

Despite its name, in order to compute the Bayesian Information Criterion it is only required to compute the likelihood of the model under analysis. The formulation of the BIC presented earlier is in fact an approximation that stems out from the posterior probability of selecting a particular model drawn from a group of submodels. The posterior probability

associated to the selection of the model $\mathcal{M}$, with $\boldsymbol{\theta}$ being its parameter vector, is, through Bayes' Theorem:

$$\pi(\mathcal{M}_S|\boldsymbol{y}) = \frac{\pi(\mathcal{M}_S)\int_{\Theta_S} f(\boldsymbol{y}|\mathcal{M}_S, \boldsymbol{\theta}_S)\pi(\boldsymbol{\theta}_S|\mathcal{M}_S)d\boldsymbol{\theta}_s}{f(\boldsymbol{y})},$$

where Θ_S is the parameter space to which $\boldsymbol{\theta}_S$ belongs and $f(\boldsymbol{y})$ is the unconditional likelihood of the data. When that particular model can be chosen from p different models, its selection probability can be computed as:

$$f(\boldsymbol{y}) = \sum_{s=1}^{p} \pi(\mathcal{M}_S) \int_{\Theta_s} f(\boldsymbol{y}|\mathcal{M}_{,s}\boldsymbol{\theta}_S)\pi(\boldsymbol{\theta}_S|\mathcal{M}_S)d\boldsymbol{\theta}_S = \sum_{s=1}^{p} \pi(\mathcal{M}_S)\,\gamma_S(\boldsymbol{y}).$$

When comparing across models, the prior $\pi(\mathcal{M}_S)$ is not important, as we can set it to be constant. The crucial part is then to obtain an approximation for the integral $\gamma_S(\boldsymbol{y})$. This quantity, when taken in logarithm and multiplied by 2, is defined as the exact BIC score of the model built with the covariates belonging to the subset S. Normally exact BIC scores are difficult to compute even when the same prior is chosen across all models. In order to reach a formulation which is easier to compute, one can use Laplace approximation, and it is possible to demonstrate that:

$$\text{BIC}_S^{exact} = 2\text{log-likelihood}_{max}(\mathcal{M}_S) - (\log n)\dim(\mathcal{M}_S) + \mathcal{O}(1),$$

which is the formula we have given at the start of this section, once the non dominant terms in $\mathcal{O}(1)$ are ignored. Eventually, the posterior probability of each model can be estimated using

$$\pi(\mathcal{M}_S|\boldsymbol{y}) = \frac{\pi(M_S)e^{\frac{1}{2}BIC_S}}{\sum_{s'=1}^{p}\pi(M'_S)e^{\frac{1}{2}BIC_{S'}}},$$

obtaining the relative merits of each one of the model during the process.

With p covariates, it is possible to generate 2^p possible models. If one wants to perform an *exhaustive search*, which consists in assigning each one of them an AIC or BIC score and then selecting the highest one, computational complexity problems might be encountered relatively easily, as p grows. A possible path to choose in order to avoid computational problems is to use a *step-wise* algorithm, such as forward or backward selection. The former method starts by fitting the null model and then proceeds by adding one covariate at each step, selecting the one the maximizes the increase in AIC or BIC score. When adding a variable does not increase the score of the model anymore, the algorithm stops and then the model selected is the one fit at that step. The backward selection algorithm is symmetric

to the one just shown, it starts from the full model and then performs covariate selection by removing one variable at each step, stopping when after this operation, the score does not increase. Note that both algorithms select then from p model at the first step, then from $p-1$, $p-2$ and so on, hence the maximum number of models that can be fit by these algorithms is then $\frac{p(p+1)}{2}$. The forward selection has to fit this number of models when the true model is the full model while backward selection, by contrast, has to fit the same number of models when the true model is the null one. Even in this extreme case, however, the computational complexity decreases from being exponential $\mathcal{O}(2^p)$ to polynomial $\mathcal{O}(p^2)$.

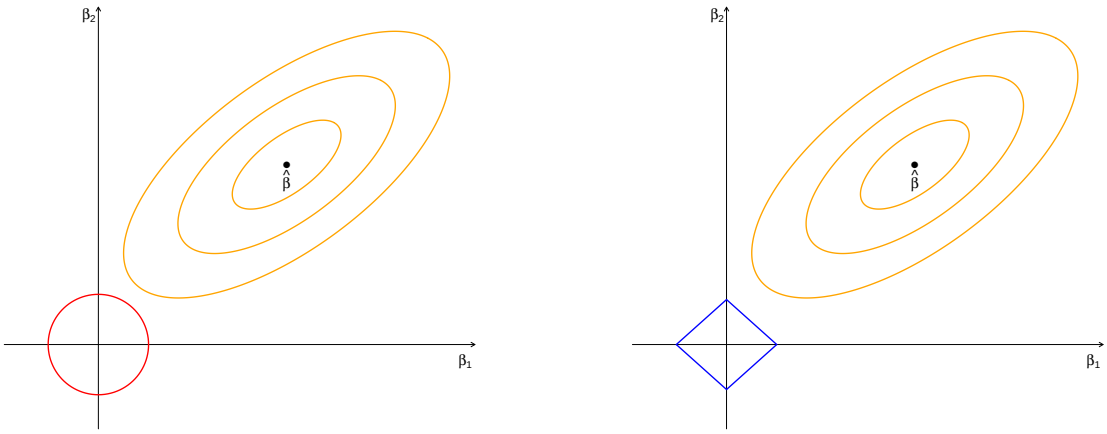
3.1.2 Lasso method

Another way to perform model selection could be using a penalized regression method different from ridge regression, which is *lasso* regression, introduced by Tibshirani [1996]. The term "lasso" is an acronym for "least absolute shrinkage and selection operator" and describes what behaviour this particular estimator has: the lasso estimator can introduce shrinkage into the estimates of the regression coefficients and variable selection at the same time by using the L1 norm as a penalty. Using the same assumptions as in Chapter 2 for the elements of the linear model, the lasso estimates are then obtained as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\lambda}^{lasso} &= \arg\min_{\boldsymbol{\beta}}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^{\top}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})+\lambda||\boldsymbol{\beta}||_1 \\ &= \arg\min_{\boldsymbol{\beta}}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^{\top}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sum_{i=1}^{p}|\beta_i|.\end{aligned}$$

The penalty parameter, λ, has the same behaviour in the lasso estimator as it has in the ridge estimator: λ controls how much weight is given to the penalty present in the constrained optimization problem, hence larger values of λ introduce a shrinkage into the estimates of $\boldsymbol{\beta}$. The use of the L1 penalty, on one hand gives the lasso the possibility to find exact zeroes in its solution, thus selecting variables, but on the other hand makes the optimization problem no longer linear in $\boldsymbol{\beta}$. The latter conclusion implies that the constrained optimization has not an enclosed form solution and so the lasso estimator must be computed solving a quadratic programming problem. Efficient algorithms are nonetheless available to solve this type of problems at the same computing cost as for ridge regression, hence the lack of an enclosed form solution does not affect the practical use of lasso.

As with ridge regression, fine-tuning of the penalty parameter, λ, is required in order to yield estimates which are set to have a lower prediction error. With the lack on an enclosed



(a) (b)

Figure 3.1: Geometrical interpretation of penalized regression. On the left the ridge penalization has smooth borders, hence permitting a solution in a point with both coefficient greater than zero. On the right side the lasso penalization forms a region with sharp corners, permitting solutions with one coefficient set to zero.

form solution for the lasso estimator of $\boldsymbol{\beta}$, it is not possible, in this framework, to carry out formulations of the mean squared which are linear in respect to the tuning parameter. If one wants to performing fine-tuning by minimizing the expected estimation error, as we proposed for ridge regression, lasso mean squared error has to be estimated first, for example by using bootstrapping techniques, as suggested by Hellton and Hjort [2018]. As previously mentioned, fine-tuning can be performed avoiding these problems with the use of cross-validation. In addition, the two extreme values that the tuning parameter, λ, can assume have the same effect to the lasso estimates as they have to the ridge ones. Setting λ equal to zero would in fact remove the effect of the penalization present in the optimization problem, thus yielding the ordinary least squares estimates. If larger value of λ are taken, lasso will introduce shrinkage into the estimates, bringing all of them to exactly zero.

The real difference between ridge and lasso regression lies in what happens in between those two situations. As we presented earlier, ridge estimates are pushed towards zero as the tuning parameter, λ, increases and they will all reach that exact value when λ tends to infinity. Lasso estimates, on the other hand, give solutions which are exactly zero even for

defined values of the tuning parameter, λ. For certain values of the tuning parameter, then, some coefficients will be null while the others will show shrinkage in their estimates. Thanks to this property, the lasso operates a continuous variable selection, which depends from λ, and hence $\hat{\boldsymbol{\beta}}_{\lambda}^{lasso}$ may contain several zeroes, becoming a sparse vector. To give a glimpse on how this could happen, we can recall that both ridge and lasso estimates are solutions of constrained optimization problems that can be geometrically interpreted. If we analyse the simple case in which $p = 2$ the ridge constraint becomes $\beta_1^2 + \beta_2^2 < c$ which correspond to the circular region in the plain in which the axis represent the two coefficients, whereas the lasso constraint $|\beta_1| + |\beta_2| < c$ draws a diamond shaped region with sharp edges, as it is possible to see in Figure 3.1a and Figure 3.1b, respectively. With c sufficiently large, these regions may contain the ordinary least squares solution $\hat{\boldsymbol{\beta}}$ but if this does not happen, we have to move away from that solution and see what happens when we meet the constraint regions. The contours around the ordinary least squares solution $\hat{\boldsymbol{\beta}}$ represent the regions of constant residual sum of squares and the solution of the constrained optimization problem is then found on the intersection between one of those contours and the constraint region. The ridge penalty has a boundary with no sharp edges, so it will almost never meet one of the contours in an interception with an axis, in which a coefficient is necessarily null. The sharp edges of the lasso penalty, make the latter to have the opposite behaviour with possible solutions in points where some coefficients are exactly zero.

Another trait that lasso regression shares with ridge regression is that its solution can be interpreted as a Bayesian posterior estimate, given the right prior. Using Bayes' Theorem, we can state the posterior density of a parameter is proportional to the prior and the likelihood of the data, as we did in Equation (2.11):

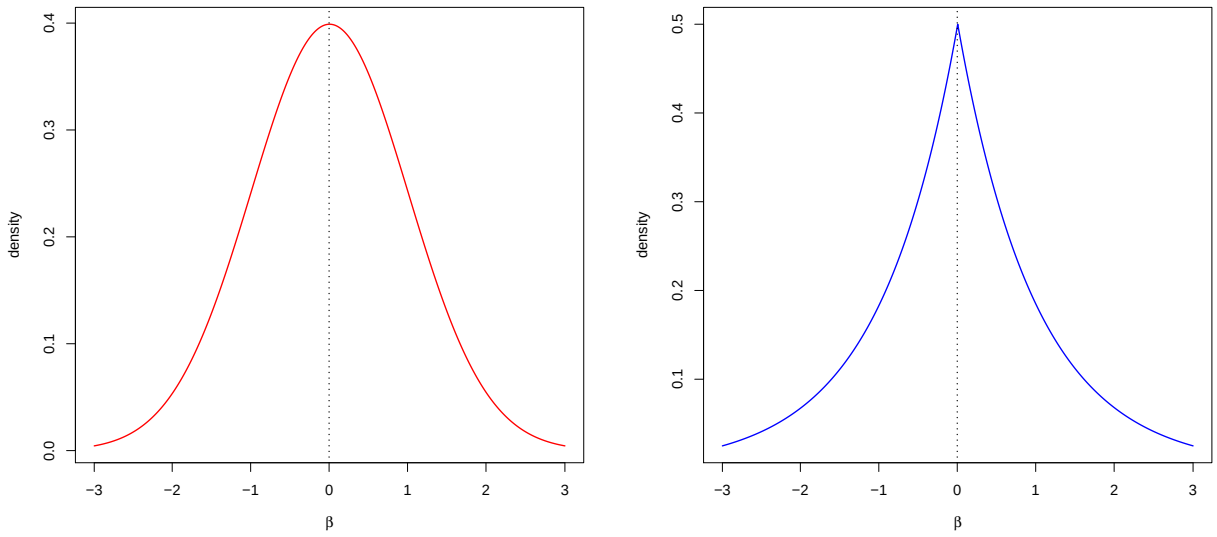
$$\pi(\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}) \propto \pi(\boldsymbol{\beta}) f(\boldsymbol{y}, \mathbf{X}|\boldsymbol{\beta}).$$

Starting form this framework, we have already shown that using a normal prior for $\boldsymbol{\beta}$ the posterior mean solution is in fact the ridge estimator. In order to obtain the lasso estimator as a Bayesian estimate, the prior of choice would have to be a double exponential with mean zero and scale parameter a function of λ. Hence, by taking such a prior:

$$\pi(\boldsymbol{\beta}) = \frac{\lambda}{2} \sum_{i=1}^{p} |\beta_i|,$$

the posterior distribution of the coefficients is:

$$\pi(\boldsymbol{\beta}|\boldsymbol{y}, \mathbf{X}) \propto \exp\left\{ -\frac{1}{2\sigma^2} \left[(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} |\beta_i| \right] \right\},$$



(a) (b)

Figure 3.2: Priors in ridge and lasso regression. Both priors are centered in 0 and have standard deviation 1. In (a) the normal prior suggests how the ridge estimates are spread around zero whereas, in (b), the lasso prior shows that in lasso regression estimates can reach the null value.

which posterior *mode* coincides with the lasso estimator. As we saw in the ridge case, it is then possible to estimate λ by using a maximum marginal likelihood method, integrating this posterior over $\boldsymbol{\beta}$. It is always possible to solve such integral numerically but it can be shown that in the particular case in which the matrix $\mathbf{X}^\top\mathbf{X}$ is diagonal the maximum marginal likelihood procedure yields:

$$\hat{\lambda}^{mml} \approx \frac{p}{\sum_{i=1}^{p} \hat{\beta}_i},$$

with $\hat{\beta}_i i = 1, \ldots, p$ being the ordinary least squares estimates of the coefficients of the $\boldsymbol{\beta}$ vector.

Referring to Figure 3.2, the shape of the priors of the two estimators show really well the differences in how the estimates are conveyed. The steepness around zero of the double exponential prior shows that *a priori* the lasso is expected to estimate multiple coefficient as null values whether the ridge is expected to yield to estimates randomly spread around zero.

3.1.3 The focussed information criterion

The techniques we have shown previously aimed at the selection of a model which could be deemed to be "best" for a wide purpose. We saw first that AIC and BIC select a model when it finds a balance between complexity and fit, and that these two criteria have some downsides: the AIC tends to be generous and has difficulty in selecting the true model when sample size increases whereas the BIC is more parsimonious and hence this behaviour makes the criterion to select models that might be too simple in some cases. Then, we saw that it is possible to perform model selection using a penalized linear regression model, which is the lasso. By changing the penalty from the L2 norm, as seen in ridge regression, into the L1 norm, increasing values of the tuning parameter, λ, help the method to produce sparse estimates of $\boldsymbol{\beta}$, thus operating variable selection indirectly. Since the model selection procedure is linked with the choice of the value of the tuning parameter, the "best" model selected by lasso might be the model that yields the lowest prediction error, if the cross-validation technique is used to assess that choice.

In some analyses, however, there might be the need to answer specific questions about some parameters of interest. Even in a simple regression framework, multiple questions might arise: we can be interested in estimating some coefficients because of a priori importance of the associated covariates, or we can be interested in yielding the best predictions of particular units, or, other times, it might be important to give an estimate of the mean response of the model. When performing model selection with the techniques previously presented, a single model will be selected by each one of them to answer to all of these different questions. On the other hand, one may think that in order to properly answer those different questions, different models are inevitably needed.

This is the main idea behind the *focussed information criterion*, also referred to as FIC, presented by Claeskens and Hjort [2008]. If the attention of the analysis is pointed towards a particular *focus*, which can be a parameter, μ, all possible models can be assigned a score in terms of how accurately these models estimate the quantity of interest μ. The FIC score of the model $\mathcal{M}_S$ is then defined to be equal to the mean squared error of its estimate of μ:

$$\text{FIC}(\mu_S) = \text{MSE}[\hat{\mu}_S] = \mathbb{E}[(\hat{\mu}_S - \mu_{true})^2].$$

The original FIC formulation proposed by Claeskens and Hjort [2008] is derived for the maximum likelihood estimator of μ_S. Its mean squared error can be derived under some mild distribution assumptions and can be written in its bias-variance decomposition. The focus μ_S is considered having a distribution depending from two vectors $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$. The former

vector contains p protected parameters that are included in every subset and constitutes the *narrow* model while the vector γ contains q parameters that might be added to the narrow model, when all of these parameters are added to the final model, the full model is chosen. Basically the search is limited between 2^q models ranging from the narrow model to the full model, when all the parameters in q are selected. Since the FIC score is indeed a mean squared error, the idea behind the criterion is to compute such quantities from (possibly) each one of the available subsets and then select the one that yields the lowest value.

Every model, indexed by S, will have a Fisher information matrix

$$\mathbf{J}_S = \begin{bmatrix} \mathbf{J}_{00,S} & \mathbf{J}_{01,S} \\ \mathbf{J}_{10,S} & \mathbf{J}_{11,S} \end{bmatrix},$$

where $\mathbf{J}_{00}$ is the $p \times p$ block referring to the parameters contained in the narrow model and $\mathbf{J}_{11,S}$ is then the $|S| \times |S|$ block referring to the $|S|$ additional parameters added to the narrow model taken from the vector γ. We can then define the variance of the narrow model as

$$\tau_0^2 = (\frac{\partial \mu}{\partial \boldsymbol{\theta}})^\top \mathbf{J}_{00}^{-1} (\frac{\partial \mu}{\partial \boldsymbol{\theta}}).$$

By assuming that the mean squared error of the estimated γ vector in the wide model is a normal with mean $\boldsymbol{\delta}$ and variance $\mathbf{J}_{11}^{-1}$, we can define the vector $\boldsymbol{\omega}$ as

$$\boldsymbol{\omega} = \mathbf{J}_{10}\mathbf{J}_{00}^{-1}(\frac{\partial \mu}{\partial \boldsymbol{\theta}}) - (\frac{\partial \mu}{\partial \boldsymbol{\gamma}}).$$

The bias of the narrow model would then be $\boldsymbol{\omega}^\top \boldsymbol{\delta}$. This vector has great importance into the FIC line of thinking. Since it contains the partial derivatives of the parameter μ, this is the element that characterizes the choice of the model. When changing the focus, this vector will be surely modified hence making the FIC able to select a different model for a different focus.

Finally, by using a suitable projection matrix π_S, formed by the rows of the identity matrix of size q that refers to the covariates present in S, every $\boldsymbol{\gamma}_S$ would then have variance equal to $\pi_S^\top \mathbf{J}_{11}^{-1} \pi_S$. Such matrix is a $q \times q$ identity matrix with rows and columns drawn by J_{11}^{-1} when belonging to the particular subset S. We can now present the original FIC formulation as

$$\begin{aligned} \text{MSE}[\hat{\mu}_S] = &\tau_0^2 + \boldsymbol{\omega}^\top \pi_S^\top \mathbf{J}_{11,S}^{-1} \pi_S \boldsymbol{\omega} \\ &+ \boldsymbol{\omega}^\top (\mathbf{I}_q - \pi_S^\top \mathbf{J}_{11,S}^{-1} \pi_S \mathbf{J}_{11}) \boldsymbol{\delta}^\top \boldsymbol{\delta} (\mathbf{I}_q - \pi_S^\top \mathbf{J}_{11,S}^{-1} \pi_S \mathbf{J}_{11})^\top \boldsymbol{\omega}. \end{aligned}$$

When moving from the narrow model to the wide model the projection matrix pi_S would select more and more elements from $\mathbf{J}_{11}^{-1}$, thus making the variance of the MSE of the relative subset larger and its bias smaller. Such projection matrix will degenerate into the $\mathbf{I}_q$ matrix when the MSE of the wide model is estimated yielding to the largest variance possible $\tau_0^2 + \boldsymbol{\omega}^\top \mathbf{J}_{11}^{-1} \boldsymbol{\omega}$ and to null bias. Hence, following the reasoning of the bias-variance trade-off there might be a model that has the right balance of the two quantities and yields to the least possible error while estimating the parameter μ.

Finally, the main result of the FIC methodology is that is it possible to perform model selection once the risk function $\mathbb{E}[(\hat{\mu}_S - \mu_{true})^2]$ is estimated given the parameter, or set of parameters of interest. Spawning from such conclusion, is then possible to move into the FIC methodology by analysing how such risk function behaves in a set of candidate models.

Another modification that could be performed refers to relaxing the focussed framework in order to find an estimator for a set of parameters averaging the risk functions of the single parameters, or starting from a global risk function that refers to all the parameters of interest simultaneously. When using those risk functions as a criterion to select submodels, it is used an *average-FIC* approach, or AFIC approach.

3.2 An average focussed information criterion for ridge regression

As we saw in Chapter 2, ridge regression is a penalized regression model that keeps all covariates while estimating the regression coefficients present in $\boldsymbol{\beta}$. Differently from what happens with lasso, ridge regression tends to keep all covariates, as the ridge penalization does not permit to have sparse estimates of $\boldsymbol{\beta}$. We might want then to build a framework in which ridge regression could operate model selection.

Stemming from the conclusions drawn regarding the FIC methodology, it is possible to use the expected estimation error of the vector $\boldsymbol{\beta}$ as a risk function to minimize over different subsets, in order to perform model selection with an AFIC criterion. The ridge estimator obtained on the subset S is then a column vector of dimensions $s \times 1$, hence it is not possible to compare it directly with the true $\boldsymbol{\beta}$ vector, which is a $p \times 1$ column vector. To solve the problem, note that it is always possible to define a $s \times p$ projection matrix π_S which permits to move between the two dimensions s and p. On this regard, when applied to the design matrix $\mathbf{X}$, the projection matrix selects the s variables forming the subset S which

we want to use, namely: $\mathbf{X}\pi_S^\top = \mathbf{X}_S$, when applied to the true vector $\boldsymbol{\beta}$ it will select the true coefficients associated with the variables in S. Otherwise, when it is needed to expand the number of dimensions from s to p, the projection matrix can be still used. The object $\pi_S^\top \hat{\boldsymbol{\beta}}_S = \hat{\boldsymbol{\beta}}_S^*$ is a vector of dimensions $p \times 1$ and has zeroes placed in the positions occupied by the covariates existing in $\mathbf{X}$ but not selected in $\mathbf{X}_S$.

Using this object is then possible to compute the mean squared error for the coefficients of the ridge regression model built using $\mathbf{X}_S$ as:

$$\text{MSE}[\hat{\boldsymbol{\beta}}_{\lambda,S}^*] = \mathbb{E}[(\hat{\boldsymbol{\beta}}_{\lambda,S}^* - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}}_{\lambda,S}^* - \boldsymbol{\beta})].$$

The parameters in $\hat{\boldsymbol{\beta}}_{\lambda,S}$, however, do not depend only from the subset in analysis S, but from the tuning parameter λ as well.

Using one of those two formulations the mean squared error as an AFIC criterion to minimize, the model selection method presented will select the subset that yields the lowest AFIC score, after a joint minimization over the tuning parameter, λ, and over all the subsets, S.

3.2.1 Mean squared error minimization

Before showing how to construct and minimize the mean squared error expression under this new framework, note that the expected value and the variance of the ridge estimator show some variations. Let S be the subset chosen and $\mathbf{X}_S$ its relative design matrix, with cardinality $|S| = s$, it is always possible to define the ridge estimator formed by using the covariates of that particular subset as $\hat{\boldsymbol{\beta}}_{\lambda,S} = \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S$. The expected value of the ordinary least squares estimator built with design matrix $\mathbf{X}_S$ is

$$\mathbb{E}[\hat{\boldsymbol{\beta}}_S] = \mathbb{E}[(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \boldsymbol{y}] = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}] = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X}\boldsymbol{\beta},$$

whereas its variance is

$$\mathbb{V}[\hat{\boldsymbol{\beta}}_S] = (\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top]\mathbf{X}_S(\mathbf{X}_S^\top \mathbf{X}_S)^{-1} = \sigma^2(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}.$$

Now it is possible to show that in this particular framework the mean squared estimation error of the ridge estimator relative to the particular subset S can be formulated as

$$\begin{aligned}
\text{MSE}[\hat{\boldsymbol{\beta}}_{\lambda,S}^*] &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_{\lambda,S}^* - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}}_{\lambda,S}^* - \boldsymbol{\beta})], \\
&= \mathbb{E}[(\pi_S^\top \hat{\boldsymbol{\beta}}_{\lambda,S} - \boldsymbol{\beta})^\top (\pi_S^\top \hat{\boldsymbol{\beta}}_{\lambda,S} - \boldsymbol{\beta})], \\
&= \mathbb{E}[(\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta})^\top (\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta})],
\end{aligned}$$

using the bias-variance decomposition seen in Equation (2.9):

$$
\begin{aligned}
&= \mathbb{E}[(\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S])^\top(\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S])] \\
&\quad + (\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S] - \boldsymbol{\beta})^\top(\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S] - \boldsymbol{\beta}), \\
&= \mathbb{E}[(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])^\top \mathbf{W}_{\lambda,S}^\top \pi_S \pi_S^\top \mathbf{W}_{\lambda,S}(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])] \\
&\quad + \boldsymbol{\beta}^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]\boldsymbol{\beta},
\end{aligned}
$$

it is possible to show that $\pi_S \pi_S^\top = \mathbf{I}_s$ and hence we obtain:

$$
\begin{aligned}
&= \mathbb{E}[(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])^\top \mathbf{W}_{\lambda,S}^\top \mathbf{W}_{\lambda,S}(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])] \\
&\quad + \boldsymbol{\beta}^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]\boldsymbol{\beta}, \\
&= \sigma^2 \mathrm{tr}[\mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{W}_{\lambda,S}^\top] \\
&\quad + \boldsymbol{\beta}^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]^\top[\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p]\boldsymbol{\beta}.
\end{aligned}
$$

With this formulation, it is possible to show why the mean squared error can be used as an averaged risk function over the parameters that has a similar behaviour as the FIC criterion. In fact, when a small subset is chosen the variance part of the decomposition is smaller while its bias is inevitably larger. In the extreme case when no covariates are selected, $S = \emptyset$, the variance term is null and the bias term reaches is maximum value. Also in this case then, we expect to find the true model when the two quantities reach a balance and hence give the minimum mean squared error among all subsets.

As we have already seen in Chapter 2, another risk function that might be used is represented by the expected prediction error. Following almost the same steps of the formulation we just reported, it is possible to show that

$$
\begin{aligned}
\mathrm{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S}] &= \mathbb{E}[(\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S} - \mathbf{X}\boldsymbol{\beta})], \qquad (3.1) \\
&= \mathbb{E}[(\mathbf{X}\pi_S^\top \hat{\boldsymbol{\beta}}_{\lambda,S} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{X}\pi_S^\top \hat{\boldsymbol{\beta}}_{\lambda,S} - \mathbf{X}\boldsymbol{\beta})], \\
&= \mathbb{E}[(\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \mathbf{X}\boldsymbol{\beta})], \\
&= \mathbb{E}[(\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S])^\top((\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\hat{\boldsymbol{\beta}}_S - \mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S])] \\
&\quad + (\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S] - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}\mathbb{E}[\hat{\boldsymbol{\beta}}_S] - \mathbf{X}\boldsymbol{\beta}), \\
&= \mathbb{E}[(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])^\top \mathbf{W}_{\lambda,S}^\top \pi_S \mathbf{X}^\top \mathbf{X} \pi_S^\top \mathbf{W}_{\lambda,S}(\hat{\boldsymbol{\beta}}_S - \mathbb{E}[\hat{\boldsymbol{\beta}}_S])] \\
&\quad + \boldsymbol{\beta}^\top[\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{X}]^\top[\mathbf{X}\pi_S^\top \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top \mathbf{X} - \mathbf{X}]\boldsymbol{\beta}, \\
&= \sigma^2 \mathrm{tr}[\mathbf{X}_S \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{W}_{\lambda,S}^\top \mathbf{X}_S^\top] \\
&\quad + \boldsymbol{\beta}^\top \mathbf{X}^\top[\mathbf{X}_S \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top - \mathbf{I}_n]^\top[\mathbf{X}_S \mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top - \mathbf{I}_n]\mathbf{X}\boldsymbol{\beta}.
\end{aligned}
$$

More formally, it is now possible to define the *ridges procedure*, which is our AFIC based model selection method:

Definition 3.2.1. *The ridges procedure selects the subset S and the value of the tuning parameter that minimize the AFIC criterion chosen among all subsets, substituting the unknown quantities with suitable plug-in estimates*

$$\{\hat{\lambda}_{\hat{S}}, \hat{S}\} = \arg\min_{\lambda,S}\{\widehat{\mathrm{MSE}}_{\lambda,\,S,\,\tilde{\boldsymbol{\beta}},\,\tilde{\sigma}^2}\},$$

where $\hat{\lambda}_S$ can be interpreted as an estimate of the oracle tuning parameter associated to the subset S

$$\hat{\lambda}_S = \arg\min_{\lambda}\{\widehat{\mathrm{MSE}}_{\lambda,\,S,\,\tilde{\boldsymbol{\beta}},\,\tilde{\sigma}^2}\}.$$

The results of the ridges procedure can be used to compute the *ridges estimator*, built on the subset deemed to be the final model, and using the estimate of its oracle tuning parameter

$$\hat{\boldsymbol{\beta}}_{\lambda,S} = (\mathbf{X}_{\hat{S}}^\top \mathbf{X}_{\hat{S}} + \hat{\lambda}_{\hat{S}}\mathbf{I}_s)^{-1}\mathbf{X}_{\hat{S}}^\top \boldsymbol{y}.$$

As a concluding remark, it is possible to note that the joint minimization over the subsets and the tuning parameter can be also performed sequentially. All the oracle tuning parameters of all subset are estimated first and plugged into the AFIC criterion formulation chosen, the subset which yields to the lowest value of the criterion after this step is then selected.

3.2.2 Singular value decomposition of the mean squared error

To avoid the computational problems that arise with the inversion of full matrices, a reduced form of the criteria that could be computed faster is certainly needed. Also in this case the solution proposed revolves around the use of the singular value decomposition of the design matrix involved into the formulation. In this case the matrix that will be decomposed in this fashion will be the design matrix associated to a particular subset, $\mathbf{X}_S$.

Starting from the expected estimation error criterion, its variance part recalls what we presented in Equation (2.12) but involving terms relative to a particular subset S. By applying the singular value decomposition on $\mathbf{X}_S$, it is then possible to express such variance as

$$\mathbb{V}\big(\mathrm{MSE}[\hat{\boldsymbol{\beta}}^*_{\lambda,S}]\big) = \tilde{\sigma}^2\mathrm{tr}[\mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{W}_{\lambda,S}^\top] = \tilde{\sigma}^2\sum_{i\in S}\left(\frac{d_{S,i}^2}{d_{S,i}^2+\lambda}\right)^2.$$

The squared bias part can be rewritten defining the $s \times n$ $\mathbf{F}_S$ matrix as $\mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top$ (substituting in the plug-in estimates, and recalling that $\pi_S \pi_S^\top = \mathbf{I}_s$):

The new formulation of the squared bias part of the criterion starts by noting that the $s \times n$ matrix formed by this matrix multiplication $\mathbf{W}_{\lambda,S}(\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\mathbf{X}_S^\top$ can be reduced to $\mathbf{V}_S \mathbf{D}_S (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1} \mathbf{U}_S^\top$. Substituting this into the bias term and defining $\pi_S \tilde{\boldsymbol{\beta}} = \tilde{\boldsymbol{\beta}}_S$, which is the partition of the original $\tilde{\boldsymbol{\beta}}$ that comprises the coefficients associated with the covariates belonging to the subset S, we obtain:

$$\begin{aligned}\text{Bias}^2\big(\text{MSE}[\hat{\boldsymbol{\beta}}^*_{\lambda,S}]\big) = &\tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{U}_S \mathbf{D}_S^2 (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-2} \mathbf{U}_S^\top \mathbf{X} \tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\beta}}^\top \tilde{\boldsymbol{\beta}} \\ &- \tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{U}_S \mathbf{D}_S (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1} \mathbf{V}_S^\top \tilde{\boldsymbol{\beta}}_S \\ &- \tilde{\boldsymbol{\beta}}_S^\top \mathbf{V}_S \mathbf{D}_S (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1} \mathbf{U}_S^\top \mathbf{X} \tilde{\boldsymbol{\beta}},\end{aligned}$$

now defining the two $s \times 1$ vectors $\mathbf{U}_S^\top \mathbf{X} \tilde{\boldsymbol{\beta}} = \boldsymbol{\xi}_S$ and $\mathbf{V}_S^\top \tilde{\boldsymbol{\beta}}_S = \boldsymbol{\nu}_S$

$$\begin{aligned}= &\boldsymbol{\xi}_S^\top \mathbf{D}_S^2 (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-2} \boldsymbol{\xi}_S + \tilde{\boldsymbol{\beta}}^\top \tilde{\boldsymbol{\beta}} \\ &- \boldsymbol{\xi}_S^\top \mathbf{D}_S (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1} \boldsymbol{\nu}_S \\ &- \boldsymbol{\nu}_S^\top \mathbf{D}_S (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1} \boldsymbol{\xi}_S.\end{aligned}$$

Every term in this expression is a multiplication of three terms, which are respectively a vector, a diagonal matrix, and then another vector, with $\tilde{\boldsymbol{\beta}}^\top \tilde{\boldsymbol{\beta}}$ being interpreted as $\tilde{\boldsymbol{\beta}}^\top \mathbf{I}_p \tilde{\boldsymbol{\beta}}$. As we saw in Chapter 2, this type of matrix multiplications reduce the computational costs of a fairly amount, when reduced to the sums of multiplications of the corresponding elements present in the two vectors and on the diagonal of the diagonal matrix. Noting that the diagonal matrices involved have ith element on their diagonal

$$\begin{aligned}\{\mathbf{D}_S^2 (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-2}\}_{ii} &= \left(\frac{d_{S,i}}{d_{S,i}^2 + \lambda}\right)^2 \quad \text{for } i = 1, \ldots, s, \\ \{\mathbf{D}_S^2 (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-1}\}_{ii} &= \frac{d_{S,i}^2}{d_{S,i}^2 + \lambda} \quad \text{for } i = 1, \ldots, s,\end{aligned}$$

and that the last two terms in the squared bias formulation are symmetric, it is then possible to reduce the expected estimation error criterion to:

$$\text{MSE}[\hat{\boldsymbol{\beta}}^*_{\lambda,S}] = \tilde{\sigma}^2 \sum_{i=1}^{s} \left(\frac{d_{S,i}}{d_{S,i}^2 + \lambda}\right)^2 + \sum_{i=1}^{s} \left(\frac{d_{S,i}\, \xi_{S,i}}{d_{S,i}^2 + \lambda}\right)^2 - 2 \sum_{i=1}^{s} \frac{d_{S,i}\, \xi_{S,i}\, \nu_{S,i}}{d_{S,i}^2 + \lambda} + \sum_{j=1}^{p} \tilde{\beta}_j^2.$$

The notation S indicates the elements that depend from the specific subset S, having cardinality $|S| = s$.

The expected prediction error criterion can also be rewritten in a easier to compute fashion using the same ideas. As we saw earlier, its variance also recalls the decomposition presented in Equation 2.13, involving only matrices that depend on S

$$\mathbb{V}\big(\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S}]\big) = \sigma^2 \text{tr}[\mathbf{X}_S \mathbf{W}_{\lambda,S} (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{W}_{\lambda,S}^\top \mathbf{X}_S^\top] = \tilde{\sigma}^2 \sum_{i=1}^{s} \left(\frac{d_{S,i}^2}{d_{S,i}^2 + \lambda} \right)^2.$$

The squared bias part of the expected prediction error criterion can be reduced by first manipulating the $n \times n$ matrix formed by the product $\mathbf{X}_S \mathbf{W}_{\lambda,S} (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top$:

$$\begin{aligned}
\mathbf{X}_S \mathbf{W}_{\lambda,S} (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top &= \mathbf{U}_S \mathbf{D}_S \mathbf{V}_S^\top \mathbf{V}_S (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-1} \mathbf{V}_S \mathbf{V}_S^\top \mathbf{D}_S \mathbf{U}_S^\top, \\
&= \mathbf{U}_S \mathbf{D}_S^2 (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-1} \mathbf{U}_S^\top.
\end{aligned}$$

Inserting this decomposition into the squared bias term and recalling the $\boldsymbol{\xi}_S$ vector defined before we obtain

$$\begin{aligned}
\text{Bias}^2\big(\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S}]\big) &= \tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{U}_S^\top \mathbf{D}_S^4 (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-2} \mathbf{U}_S \mathbf{X} \tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{X} \tilde{\boldsymbol{\beta}} \\
&\quad - 2 \tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{D}_S^2 (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-1} \mathbf{X} \tilde{\boldsymbol{\beta}} \\
&= \boldsymbol{\xi}_S^\top \mathbf{D}_S^4 (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-2} \boldsymbol{\xi}_S + \tilde{\boldsymbol{\beta}}^\top \mathbf{X}^\top \mathbf{X} \tilde{\boldsymbol{\beta}} - 2 \boldsymbol{\xi}_S^\top \mathbf{D}_S^2 (\mathbf{D}_S + \lambda \mathbf{I}_s)^{-1} \boldsymbol{\xi}_S.
\end{aligned}$$

This expression involves a new diagonal matrix, which has ith elements on its diagonal:

$$\{\mathbf{D}_S^4 (\mathbf{D}_S^2 + \lambda \mathbf{I}_s)^{-2}\}_{ii} = \left(\frac{d_{S,i}}{d_{S,i}^2 + \lambda} \right)^2 \quad \text{for } i \in S,$$

hence, it is possible to reduce the mean squared prediction error criterion as a sum of sums, noting that the other elements involved in its expression are either vectors or diagonal matrices

$$\text{MSE}[\mathbf{X}\hat{\boldsymbol{\beta}}^*_{\lambda,S}] = \tilde{\sigma}^2 \sum_{i=1}^{s} \left(\frac{d_{S,i}^2}{d_{S,i}^2 + \lambda} \right)^2 + \sum_{i=1}^{s} \left(\frac{d_{S,i}^2 \, \xi_{S,i}}{d_{S,i}^2 + \lambda} \right)^2 - 2 \sum_{i=1}^{s} \frac{(d_{S,i} \, \xi_{S,i})^2}{d_{S,i}^2 + \lambda} + \sum_{j=1}^{n} (\boldsymbol{x}_j^\top \tilde{\boldsymbol{\beta}})^2,$$

where $\boldsymbol{x}_j^\top$, with $j = 1, \ldots, n$ are the rows of the design matrix $\mathbf{X}$.

As a concluding remark, the double dependence from λ and S is reinforced using this formulations because it can be demonstrated that the singular values decompositions of the partitions of $\mathbf{X}$, $\mathbf{X}_S$ used for all the subsets are not the equal to the same partitions of the singular value decomposition of the design matrix $\mathbf{X}$.

3.3 A focussed information criterion for ridge regression

The two criteria presented in the previous section involved the use of a known response vector $\boldsymbol{y}$. In some cases, we might be interested in performing predictions having data with no associated real response value yet. It is possible to modify this criteria in order to allow the ridge method to operate in a setting in which the only aim is prediction of new values of the response variable. We will start by modifying the mean squared error minimization method and then we will introduce the technique termed *fridge* [Hellton and Hjort, 2018], which utilises a focussed prediction approach, tailoring the fine tuning of the tuning parameter λ for every new observation one wants to give prediction on.

A first way that could be pursued to insert this attention to prediction into ridge regression involves the use of another type of criterion, when fine-tuning of the penalty parameter, λ, using the minimization of the mean squared error approach. Hence the value of the tuning parameter, λ, is chosen to minimize the expected prediction error on new data, stored in the $m \times p$ matrix $\mathbf{X}_0$, whereas the plug-in estimates of the vector $\boldsymbol{\beta}$ and the variance of the error term σ^2 are always computed using data contained in the data matrix $\mathbf{X}$, for which are available the values of the response variable, stored in the vector $\boldsymbol{y}$. Following what we presented in Equation (2.10), such mean squared error could be computed as, in its bias-variance decomposition:

$$\begin{aligned}\text{MSE}[\mathbf{X}_0\boldsymbol{\beta}_\lambda] &= \mathbb{E}[(\mathbf{X}_0\hat{\boldsymbol{\beta}}_\lambda - \mathbf{X}_0\boldsymbol{\beta})^\top(\mathbf{X}_0\hat{\boldsymbol{\beta}}_\lambda - \mathbf{X}_0\boldsymbol{\beta})], \\ &= \sigma^2 \text{tr}[\mathbf{X}_0\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top\mathbf{X}_0^\top] + \boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_p)^\top\mathbf{X}_0^\top\mathbf{X}_0(\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta}.\end{aligned}$$

Hellton and Hjort [2018], however, suggest that in certain situation it could be fruitful to fine-tune the penalty parameter λ over the expected prediction error of the prediction of a new single unit. This focussed approach in regards to fine-tuning the penalty parameter, λ, is termed *fridge*, and can be interpreted as a special case of the criterion presented before. In the fridge case, the data matrix containing new data $\mathbf{X}_0$ is reduced to a vector, the single unit we wish to fine tuning the penalty parameter λ on:

$$\begin{aligned}\text{MSE}[\boldsymbol{x}_0\boldsymbol{\beta}_\lambda] &= \mathbb{E}[(\boldsymbol{x}_0^\top\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{x}_0^\top\boldsymbol{\beta})^2], \\ &= \sigma^2\boldsymbol{x}_0^\top\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top\boldsymbol{x}_0 + \left[\boldsymbol{x}_0^\top(\mathbf{W}_\lambda - \mathbf{I}_p)\boldsymbol{\beta}\right]^2.\end{aligned}$$

Following the minimization of this focussed criterion every new unit $\boldsymbol{x}_0$ will have a particular value of the tuning parameter λ associated with it.

More formally, in this setting we can define both the *focussed oracle tuning*

Definition 3.3.1. *The focussed oracle tuning, referring to the new observation* $\boldsymbol{x}_0$ *is*

$$\lambda_{\boldsymbol{x}_0} = \arg\min_{\lambda} \left\{ \mathrm{MSE}[\boldsymbol{x}_0^\top \boldsymbol{\beta}_\lambda] \right\},$$

when the quantities $\boldsymbol{\beta}$ *and* σ^2 *are known.*

and its estimate, using the fridge procedure

Definition 3.3.2. *The focussed oracle tuning can be estimated by the fridge, substituting the unknown quantities with suitable plug-in estimates* $\tilde{\boldsymbol{\beta}}$ *and* $\tilde{\sigma}^2$

$$\hat{\lambda}_{\boldsymbol{x}_0} = \arg\min_{\lambda} \left\{ \widehat{\mathrm{MSE}}[\boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}}_\lambda] \right\}.$$

It is possible to show, through the geometrical interpretation of the explicit solution for λ in a special case, that a focussed approach to fine-tuning has a unique property.

If we let the data matrix $\mathbf{X}$ to have orthogonal columns, it will have a covariance matrix which is diagonal, with equal entries. The $\mathbf{X}^\top \mathbf{X}$ matrix will be then equal to $m\mathbf{I}_p$, where m denotes the generic entry on the diagonal. In this case, the focussed criterion reduces to:

$$\mathrm{MSE}[\boldsymbol{x}_0 \boldsymbol{\beta}_\lambda] = \sigma^2 ||\boldsymbol{x}_0||^2 \frac{m}{(m+\lambda)^2} \mathbf{I}_p + (\boldsymbol{x}_0^\top \boldsymbol{\beta})^2 \frac{\lambda^2}{(m+\lambda)^2} \mathbf{I}_p,$$

offering an explicit solution for the minimization problem over λ and it is:

$$\lambda_{\boldsymbol{x}_0} = \frac{\sigma^2 \, ||\boldsymbol{x}_0||^2}{(\boldsymbol{x}_0^\top \boldsymbol{\beta})^2}.$$

Now, when the unknown quantities σ^2 and $\boldsymbol{\beta}$ are known, all the attention shifts to the relationship between the vector $\boldsymbol{x}_0$ containing the covariate values of the new unit and the vector $\boldsymbol{\beta}$. In order to let this relationship emerge, it is possible to interpret the dot product $\boldsymbol{x}_0^\top \boldsymbol{\beta}$ geometrically. The dot product, or internal product, of the two vectors can be in fact manipulated as:

$$\boldsymbol{x}_0^\top \boldsymbol{\beta} = ||\boldsymbol{x}_0|| \, ||\boldsymbol{\beta}|| \cos \alpha_{\boldsymbol{x}_0},$$

where $\alpha_{\boldsymbol{x}_0}$ is the angle between $\boldsymbol{x}_0$ and $\boldsymbol{\beta}$ in the covariate space. Plugging in this manipulation into the explicit solution for $\lambda_{\boldsymbol{x}_0}$, we obtain:

$$\lambda_{\boldsymbol{x}_0} = \frac{\sigma^2}{||\boldsymbol{\beta}||^2 \cos^2 \alpha_{\boldsymbol{x}_0}}. \tag{3.2}$$

The oracle value of λ for a certain $\boldsymbol{x}_0$ depends then on how close is the vector $\boldsymbol{x}_0$ to the vector $\boldsymbol{\beta}$. While using centred variables, when the prediction associated to $\boldsymbol{x}_0$ is close to the mean response zero, $\alpha_{\boldsymbol{x}_0}$ approaches zero and then the associated value of the tuning parameter, λ, tends to infinity. On the other hand, when the prediction associated $\boldsymbol{x}_0$ is farther from the mean value zero the angle grows, reducing the penalization associated to such particular unit. Hence, it is possible to note how the fridge procedure exploits the possibility of shrinkage to yield better predictions: units that are expected to have a response value close to zero receive more shrinkage than units that are expected to have a larger associated response value. Hellton and Hjort [2018] show in their publication that this approach yields to interesting results when the data shows the presence of clustering into the covariates space.

In this framework, it is possible to use the risk function associated with the single new prediction as a proper focussed information criterion in order to build a method for *personalised model selection*. This way, also model selection can be tailored on single new observation, with the aim of yielding both better predictions and having more insight on the true model behind the data. More formally, we can define this criterion as

$$\begin{aligned}\mathrm{MSE}[\boldsymbol{x}_0^\top \boldsymbol{\beta}^*_{\lambda,S}] &= \mathbb{E}[(\boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}}^*_{\lambda,S} - \boldsymbol{x}_0^\top \boldsymbol{\beta})^2], \\ &= \sigma^2 \boldsymbol{x}_{0,S}^\top \mathbf{W}_{\lambda,S} (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{W}_{\lambda,S}^\top \boldsymbol{x}_{0,S} + \left\{ \boldsymbol{x}_0^\top [\pi_S^\top \mathbf{W}_{\lambda,S} (\mathbf{X}_S^\top \mathbf{X}_S)^{-1} \mathbf{X}_S^\top \mathbf{X} - \mathbf{I}_p] \boldsymbol{\beta} \right\}^2,\end{aligned}$$

and the combination of tuning parameter and subset chosen, $\left\{\hat{\lambda}_{\hat{S}}, \hat{S}\right\}$, ought to be found through the joint minimization in the same fashion as we previously discussed. It is possible to define this new procedure, termed *fridges*.

Definition 3.3.3. *The fridges procedure selects the subset S and the value of the tuning parameter that minimize the AFIC criterion chosen among all subsets, substituting the unknown quantities with suitable plug-in estimates*

$$\left\{\hat{\lambda}_{\boldsymbol{x}_0,\hat{S}}, \hat{S}\right\} = \arg\min_{\lambda,S} \{\widehat{\mathrm{MSE}}[\boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}}^*_{\lambda,S}]\},$$

where $\hat{\lambda}_{\boldsymbol{x}_0,S}$ can be interpreted as an estimate of the oracle tuning parameter associated to the subset S

$$\hat{\lambda}_{\boldsymbol{x}_0,S} = \arg\min_{\lambda} \left\{\widehat{\mathrm{MSE}}[\boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}}^*_{\lambda,S}]\right\}.$$

Even though the potential of a method that could performed personalized covariate selection could be certainly useful, our concerns revolve around the possibility to translate

into this framework the result on the fridge tuning parameter, $\lambda_{\boldsymbol{x}_0}$, expressed in Equation 3.2. Since the estimates of this parameter are directly linked with the square length of $\boldsymbol{\beta}$, which measures the signal strength contained in it, a model selection method that stems from this framework has to be careful in its covariate choices. Excluding important variables may interact with the tuning parameter in unexpected ways, maybe yielding to value of the criterion not carrying meaningful information. Moreover, since every subset is treated as a separate model to minimize, sometimes that result can work in the opposite direction, finding values of the penalty parameter λ that minimize criteria associated to subset which do not contain meaningful covariates, hence giving them an higher rank than they should have.

As a concluding remark, also in this case we can manipulate also this criterion with the help of the singular value decomposition of the $\mathbf{X}_S$ matrix. Having replaced the data matrix $\mathbf{X}_0$ with the vector $\boldsymbol{x}_0^\top$ and recalling the definition of the vector $\boldsymbol{\xi}_S = \mathbf{V}_S\mathbf{U}_S^\top\mathbf{X}\boldsymbol{\beta}$, the reduced form of this criterion now is, after substituting $\boldsymbol{\beta}$ and σ^2 with their plug-in estimates:

$$\mathrm{MSE}[\boldsymbol{x}_0^\top\hat{\boldsymbol{\beta}}^*_{\lambda,S}] = \tilde{\sigma}^2\sum_{i=1}^{s}\left(\frac{\boldsymbol{x}_{0,S}^\top\boldsymbol{v}_{i,S}\,d_{i,S}}{d_{i,S}^2+\lambda}\right)^2 + \left[\sum_{i=1}^{s}\left(\frac{\boldsymbol{x}_{0,S}^\top\boldsymbol{v}_{i,S}\,\xi_{i,S}\,d_{i,S}}{d_{i,S}^2+\lambda}\right) - \boldsymbol{x}_0^\top\tilde{\boldsymbol{\beta}}\right]^2,$$

where $\boldsymbol{v}_{i,S}^\top$ with $i = 1,\ldots,s$ are the rows of the matrix $\mathbf{V}_S$, from the definition of singular value decomposition of the $\mathbf{X}_S$ matrix.

Chapter 4

Simulation study and application of ridges methodology

The potential of our ridges method, which was presented in Chapter 3, is assessed through a simulation study, as it represents a closed and controlled environment. Being the ridges a model selection method, the main goal of the simulation study is to prove if it is able to identify the true model behind the data. For this reason, every simulation involves the use of a sparse vector $\boldsymbol{\beta}$. By having multiple true coefficients set to exactly zero, only a part of the covariates present in $\boldsymbol{\beta}$ form the true model, and hence are expected to be selected by the ridges.

As noted in Section 3.2.1, in order to be able to select a model, our method performs a joint minimization over the possible subsets and the tuning parameter, λ. In this case, every subset will have an oracle value of the tuning parameter, λ, which depends also in this case on the unknown quantities σ^2 and $\boldsymbol{\beta}$. We expect the subset formed by the true model to be selected.

Given the particular setting, computational efficiency is highly important. Each implementation of the ridges method requires several minimizations, as it has to estimate the oracle value of λ associated with each subset. For this reason, the simplified versions of the mean squared error expression were given in Chapter 3 will now be used throughout the simulations.

4.1 Simulation study

The analysis starts with the creation of the design matrix $\mathbf{X}$ which comprises n units and p covariates, with units drawn from $\mathcal{N}(0, 3)$. Then, in order to be able to simulate the response vector $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}$, as we already did in Section 2.7.3, the error term, $\boldsymbol{\epsilon}$ is drawn from $\mathcal{N}(0, 1)$. Eventually, 2000 response vectors are simulated following this linear relationship.

The analysis is carried out principally to assess the performance of the ridges method regarding the identification of the true model, or true subset. In order to accomplish this task, every simulation involved the use of a sparse formulation of the $\boldsymbol{\beta}$ vector. For example, in one simulation setting the $\boldsymbol{\beta}$ vector used contained eight regression coefficients and was defined by

$$\boldsymbol{\beta} = (b, b, -b, -b, 0, 0, 0, 0)^\top = \begin{bmatrix} \boldsymbol{\beta}_{true} \\ \boldsymbol{\beta}_{noise} \end{bmatrix}.$$

Being $\boldsymbol{\beta}$ a sparse vector, then, it can be partitioned in two vectors, one containing only the active variables that form the true model, and the other containing the noise, or inactive, variables. Starting from this setting, we assessed the variable selection performance of the ridges by using two metrics, computed on the final models chosen in each one of the 2000 iterations. The first one is termed as *true model inclusion rate* and it is computed as the relative frequency of final models selected by the ridges that contains the whole partition $\boldsymbol{\beta}_{true}$. This metric, however, does not account for false selections, as the true model might be fully included in the final model with the addition of some noise variables. For this reason, a second metric was used, termed *true variables selection rate*, and it is computed as the relative frequency of true covariates present in the final model. Since the first metric reaches 100% when the true model, or true subset, is chosen and the second is equal to 100% when only active variables are chosen, our main hope is to find values of both metrics close to 100%. A high score in both metrics would indicate that the ridges is able to identify correctly the whole true subset, without adding many noise variables to its selections, throughout all the iterations of the simulation.

Another important aspect to take into account is the strength of the signal of the active variables. For this reason, several values of the absolute value of the active coefficients, b, were used throughout the simulations. Those values are $0.05, 0.1, 0, 5$ and 1. We expect both plug-ins and both criteria to perform better when the signal strength increases, the ridge plug-in, in particular, is expected to need less signal strength, given its tendency to shrink the estimates.

The simulations are carried out in two main settings, a low dimensional case and a high dimensional case. The two cases have different dimensions of the design matrix, with the first having 200 units and eight covariates and second having 40 units and 40 covariates. In the latter case only the ridge estimates were used as a plug-in, as the high dimensional nature of the setting makes the ordinary least squares estimates to be unstable. Another important aspect which was explored refers to the techniques which can be used in order to select the final model. The optimal way to perform subset selection ought to be through an exhaustive search. This type of search implies that all the joint minimizations of the ridges method would be performed on all of the 2^p possible subsets. As suggested in Section 3.1.1, the computational complexity is exponential, $\mathcal{O}(2^p)$, and hence performing an exhaustive search becomes rapidly infeasible. When $p > 10$, it is common practice to use other techniques. In the high dimensional setting, other techniques will be used, such as *pseudo exhaustive search*, which can be performed both by limiting the size of the subsets to explore during the search, and thus reducing the otherwise enormous list of possible models, or by randomly extracting a list of possible models which the ridges has to evaluate. Another possible solution is represented by the use of a stepwise algorithm in which at every step a variable is added, in the forward version, or removed, in the backward version, if its addition or subtraction improves the AFIC score. Since the stepwise algorithm reduces the computational complexity by a fair amount, and has a computational complexity which is polynomial, it can be used when p exceeds ten.

4.1.1 Low dimensional setting

The first setting of the simulation study is low dimensional with $n = 200$ and $p = 8$, and a $\boldsymbol{\beta}$ vector having four active covariates. Since in this case $p < 10$, the optimal subset is found through an exhaustive search among all the $2^p = 256$ possible subsets. In this setting two correlation schemes were used in order to test the behaviour of the plug-in estimates of $\boldsymbol{\beta}$ and σ^2. The two cases follow the settings we have already used in Section 2.7.3, and, more precisely, the two cases explored are:

1. $\boldsymbol{\beta} = (b, b, -b, -b, 0, 0, 0, 0)$ and $\mathbf{X}$ with $\boldsymbol{\Sigma}_{ij} = 0.2$ for $i \neq j$.
2. $\boldsymbol{\beta} = (b, b, -b, -b, 0, 0, 0, 0)$ and $\mathbf{X}$ with $\boldsymbol{\Sigma}_{ij} = 0.9$ for $i \neq j$.

The following tables report the results on the two metrics of choice in the two cases, for every value of b used. We expect the ridge plug-in to yield better results in the latter case,

	True model inclusion rate				True variables selection rate			
	Est-OLS	Est-R	Pred-OLS	Pred-R	Est-OLS	Est-R	Pred-OLS	Pred-R
$b = 0.05$	46.5%	46.3%	46.4%	46.3%	73.9%	74.0%	74.2%	74.3%
$b = 0.1$	98.9%	98.9%	98.9%	98.9%	77.8%	77.9%	77.8%	78.0%
$b = 0.5$	100.0%	100.0%	100.0%	100.0%	77.9%	78.0%	77.9%	78.0%
$b = 1$	100.0%	100.0%	100.0%	100.0%	77.9%	77.9%	77.9%	78.0%

Table 4.1: Variable selection performance in the low dimensional case with low correlation. The wording "Est" or "Pred" indicates which criterion was used in the ridges method, with the first indicating the expected estimation error and the second the expected prediction error. The wording after the dash indicates which plug-in was used, with "OLS" referring to the ordinary least squares, "R" to ridge ones.

as the presence of high correlation causes instability into OLS estimates, although they can still be computed.

The results obtained when the correlation between covariates is low show that there is little difference between the plugins and the criteria used. What dominates the selection patterns is the signal strength of the true regression coefficients. The true subset is correctly identified in its entirety only when the signal strength grows. Meanwhile, the percentage of true selections stabilises fairly quickly around 78%, meaning that, even when the true model is not fully included into the final selection, roughly four fifths of the selected variables are represented by covariates belonging to the true model.

When the correlation between the covariates is high, we obtain similar results. Surprisingly, little difference between the use of plugins or criteria was found again. Furthermore, it

	True model inclusion rate				True variables selection rate			
	Est-OLS	Est-R	Pred-OLS	Pred-R	Est-OLS	Est-R	Pred-OLS	Pred-R
$b = 0.05$	3.7%	3.7%	3.6%	3.5%	58.1%	58.2%	57.7%	57.8%
$b = 0.1$	19.5%	19.1%	19.9%	19.8%	68.6%	68.8%	68.7%	68.9%
$b = 0.5$	100.0%	100.0%	100.0%	100.0%	78.1%	78.3%	70.0%	78.1%
$b = 1$	100.0%	100.0%	100.0%	100.0%	78.1%	78.3%	78.0%	78.1%

Table 4.2: Variable selection performances in the low dimensional case with high correlation. Criteria and plug-in used are signaled following the scheme used in 4.1.

is possible to note that the signal strength dominates the selection behaviour also in this case, while the high correlation plays a role in creating more disturbance, hence a larger signal strength is required in order to stabilise the selection metrics. In this case the true selection percentage is just above the 50% when the signal is low, but it shows the same behaviour as the low correlation case, stabilising around 78%, when the signal is large enough.

4.1.2 High dimensional setting

In the high dimensional setting the dimensions of the data matrix are $n = 40$ and $p = 40$, and the data matrix is then manipulated in order to have a covariance matrix $\boldsymbol{\Sigma}$ in which $\boldsymbol{\Sigma}_{ij} = 0.8$ for $i \neq j$. In addition, given the particular settings of the problem, only the ridge plug-in was used. In a high dimensional setting, as previously stated, it is not feasible to perform an exhaustive search. For this reason, in this simulation were used other methods in order to search for the optimal subset:

1. Subset search performed pseudo exhaustive search over all the possible subsets of maximum cardinality equal to two ($\boldsymbol{\beta}$ with only two active coefficients).

2. Subset search performed through a forward stepwise algorithm ($\boldsymbol{\beta}$ with eight active coefficients).

3. Subset search performed through a pseudo exhaustive search through randomly generated subsets of size randomly drawn from a uniform ($\boldsymbol{\beta}$ with eight active coefficients).

4. Subset search performed through a pseudo exhaustive search through randomly generated subsets of size randomly drawn from a binomial ($\boldsymbol{\beta}$ with eight active coefficients).

Case 1 involved the use of the most intuitive solution: in order to avoid the computational problems due to the creation of the exhaustive list of all possible subsets, this first iteration involved a $\boldsymbol{\beta}$ vector with only few active regression coefficients. In this case it is possible to limit the search only through those models that can actually host the active partition of the regression coefficients' vector, $\boldsymbol{\beta}_{true}$, formed by only two coefficients. In Case 2 a forward stepwise algorithm was used, starting from the null model.

In the latter two cases the solution adopted was to feed the ridges method with randomly selected subsets from the exhaustive list of all possible subsets. For every application of the ridges method, 1000 random subsets were generated. In order to accomplish such task two path were chosen to generate the candidate subsets, both having the same scheme: first

the size of the subset is chosen with a random sampling from a given distribution, then the actual covariates that form the candidate subset are randomly chosen from the pool of all covariates, with an equal probability to be chosen. In Case 3 a discrete uniform distribution with support from 1 to 40 was used in the first step. This way, also all possible sizes have equal probability to be chosen. From one hand this approach does not respect the actual distribution of sizes present on the exhaustive list, but on the other gives the ridges more possibilities to choose smaller subsets, with the hope of having the most active variables as possible to be included in those subsets. In Case 4, a binomial distribution with parameters 0.5 and 40 was used. Using this distribution, the subsets proposed will have size centred around the expected value of 20, and this respects the actual distribution of sizes present on the exhaustive list. On the other hand, with 40 covariates, the amount of possible subsets containing 20 covariates is large, hence the probability to generate a candidate subset that contains a large portion of the active set is low.

Regarding the results of Case 1 reported in Table 4.3, lower values of the metrics regarding the true selections frequency were expected, given the larger number of covariates involved. Also in this case, the method reaches interesting results with a greater signal strength. Note that the high score associated to the true variables selection rate when $b = 1$ is certainly a product of the pseudo-exhaustive search method used, which utilized a priori knowledge regarding the size of $\boldsymbol{\beta}_{true}$ in order to contain the cardinality of the candidate subsets. This result can also be interpreted, in conjunction with what we showed regarding the low dimensional setting, as a justification for the use of the pseudo-exhaustive search when informations about the amount of true variables are present. This way it is possible to force the ridges to not include noise variables in its final selections. It is also possible to note that

	TMI rate		TVS rate	
	Est-R	Pred-R	Est-R	Pred-R
$b = 0.05$	0.1%	0.3%	7.3%	6.2%
$b = 0.1$	0.3%	1.3%	11.1%	11.0%
$b = 0.5$	32.7%	78.5%	60.2%	88.7%
$b = 1$	83.1%	100.0%	91.5%	100.0%

Table 4.3: Variable selection in high dimensional setting, with pseudo-exhaustive search through the subset with maximum cardinality two, and $\boldsymbol{\beta}_{true}$ containing two elements. Criteria and plug-in used are signaled following the scheme used in 4.1. Due to the high dimensional framework, only the ridge plug-in was used.

	Forward		Pseudo - U		Pseudo - B	
	Est-R	Pred-R	Est-R	Pred-R	Est-R	Pred-R
$b = 0.05$	12.8%	13.1%	34.0%	18.1%	0.0%	0.0%
$b = 0.1$	13.6%	13.5%	35.5%	19.9%	0.0%	0.0%
$b = 0.5$	49.9%	67.7%	73.4%	69.3%	0.0%	0.0%
$b = 1$	95.4%	99.9%	98.7%	96.1%	0.0%	0.0%

(a) True model inclusion rate in Cases 2, 3 and 4.

	Forward		Pseudo - U		Pseudo - B	
	Est-R	Pred-R	Est-R	Pred-R	Est-R	Pred-R
$b = 0.05$	20.3%	20.5%	20.1%	19.3%	20.2%	20.6%
$b = 0.1$	21.6%	22.2%	21.5%	20.8%	21.0%	21.2%
$b = 0.5$	33.1%	37.2%	25.2%	24.4%	28.1%	27.4%
$b = 1$	32.9%	39.8%	23.3%	25.8%	30.7%	26.8%

(b) True variables selection rate in Cases 2, 3 and 4.

Table 4.4: Simulation results in high dimensional setting, with $\boldsymbol{\beta}_{true}$ of cardinality equal to eight. Criteria and plug-in used are signaled following the scheme used in 4.1. Due to the high dimensional nature of the simulation only the ridge plug-in was used. Subset selection methods are indicated with "Forward", "Pseudo-U", referring to the pseudo-exhaustive serach with discrete uniform sampling for the sizes of the candidate subsets and "Pseudo-B", when the sizes were randomly drawn from a binomial.

in this case, the minimization of the expected prediction error yields to a faster stabilization of the selection metrics, as the signal strength increases. Hence, as a concluding remark, it can be stated that this criterion should be preferred in an high dimensional setting.

In Cases 2, 3 and 4 the $\boldsymbol{\beta}$ vector has eight true variables. Given these circumstances, the results of the simulations for these cases are presented together in Table 4.4, with Table 4.4a reporting their performance on the true model inclusion rate, and Table 4.4b reporting their performance regarding the true variables selection rate.

The results obtained by using a stepwise selection approach highlight that by not controlling the size of the subsets, it is possible to find more active covariates even when the signal strength is fairly low. The true selections frequency shows that the method tends to select also a large number of noise variables. Even though those score might seem low, they

are still better than the ones that would be scored by a forward algorithm which does not perform any model selection. When the full model is selected, in fact, it will include the true model, but it will only have a 20% of true variables. Hence, if we let this to be a baseline value, the forward algorithm passes this test. However, being the aim of model selection the identification of a model which is the closest as possible to the true model, the forward selection seems to fail in this intent. A possible modification would be to introduce also in a forward selection approach a limit on the size of the subsets it can evaluate, expecting this limitation to force the algorithm to exclude noise variables. Judging by both metrics, the criterion to be preferred when using a forward stepwise algorithm is the expected prediction error.

Performing a pseudo exhaustive search with subset sizes drawn from a discrete uniform, the values scored on the first metric are similar to the one scored by the forward stepwise algorithm. In addition, this method seems to converge more rapidly to higher values of the metric when the signal strength increases. When the sizes of the subsets are drawn from a binomial, a subset which contains the whole $\boldsymbol{\beta}_{true}$ partition is never chosen. This aspect is due to the scarcity of candidates subsets that contains such partition. Shifting the attention to the true selections frequency both methods perform similarly to the forward stepwise algorithm when the signal is low but do not show a tendency to improve their scores when the signal increases. Between the two pseudo exhaustive searches, the binomial based one has consistently higher scores, due to the fact that the candidate subsets is this case are smaller, hence they necessarily contain less inactive variables. When using one of these two latter methods, the expected estimation error has to be preferred as a criterion, as it gives higher scores on the metrics used in this study.

4.2 Application

It is now presented an application of the ridges method on a real dataset, comparing its results with other established model selection methods, chosen from the ones presented in Chapter 3. To that regard, it was chosen a contained dataset, in order to be able to discern and dissect the choices of every model selection method.

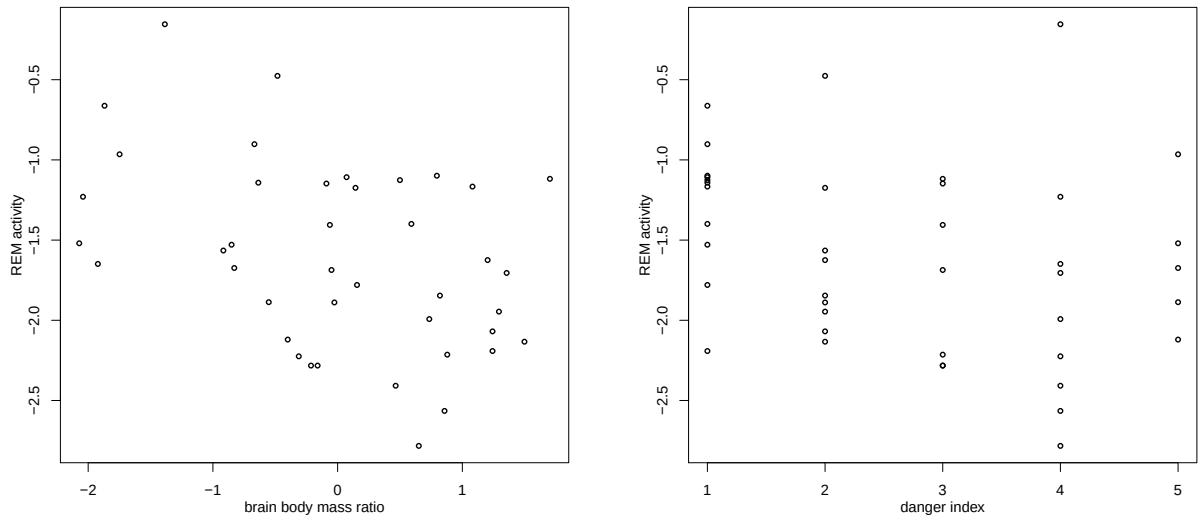
The data used comprises covariates that describe sleeping and dreaming habits of mammalians, and it is taken from [Allison and Cicchetti, 1976]. In particular, the covariates are:

- **Dreaming** (t_{REM}): hours per day of dreaming, namely REM activity during sleep.

- **Total sleep** (t_{tot}): hours per day of sleep.
- **Brain mass** (M_{br}): weight of brain (in grams) of a given mammal.
- **Body mass** (M_{bd}): weight of the body (in kilograms) of a given mammal.
- **Lifespan** (L): life expectancy in years.
- **Gestation** (G): gestation period in days.
- **Predation** (P): predation index, 1 is minimum threat of being preyed upon, 5 maximum.
- **Exposure** (E): exposure to danger during sleep index, 1 indicates a very protected sleep (in a well built den, for example), 5 indicates a large exposure to danger during sleep.
- **Danger** (D): overall danger index, 1 is minimum danger, 5 maximum, not necessarily related with predation and exposure during sleep factors.

Theories regarding sleep in mammals suggest that its main function should be repairing the brain cellular damage suffered during the awake time, as reported in Siegel [2005]. However, no quantitative theories were explored until Savage and West [2007] suggested that the amount of sleep should be correlated with brain and body mass, following a power law, hence by taking the logarithm of both quantities the relationship becomes linear. The correlation stems from the link of brain and body mass to the metabolic rates of a given mammal, with a inverse relationship. Smaller mammals have hence higher metabolic rates, experiencing more damage during their awake time and thus needing more sleep that bigger animals. Also, the amount of sleep is correlated with other physiological and ecological factor, for instance the state of danger that a mammal is subject to, both generally and during its sleep.

In regards to the part of sleep devoted to REM (Rapid Eye Movement) activity, Siegel [2005] proposes that its main function should cover the reorganization of the reparation of cellular damage that take place during sleep. In this particular phase, the brain returns to be active in regard to its metabolic activities. With the brain being active, but with no conscious control over it, the body of a mammal may move, especially in the form of rapid twitches in the extremities, hence mammals which are exposed to danger should devote less sleeping time in REM activities. Also the amount of REM sleep should be correlated with both physiological and ecological factors. However, Savage and West [2007] find that the



(a) (b)

Figure 4.1: Scatter plots of portion of REM activities during sleep against the two most important covariates. In (a) the portion of REM sleep of every mammal is plotted against their brain body weight ratio, in (b), we have the scatter plot of REM activity plotted against the danger index.

amount of REM sleep is not correlated with the body mass of a mammal, suggesting that the damage reorganizing function of this sleep phase is not dependent of the metabolic rate of the body of a particular mammal.

Given that the data contains covariates that can explain both the physiological and ecological traits of a given mammal, we explored which of those covariates could explain the amount of sleeping time devoted to REM activities for several mammalian species. We found that the amount of REM sleep is strongly correlated with the ratio of the masses of brain and body. Since the literature [Savage and West, 2007] suggested that all sleep related behaviours of mammals are linked with their metabolic rates through power laws relationship, we investigated this particular relationship:

$$\frac{t_{REM}}{t_{non-REM}} \approx \left(\frac{M_{br}}{M_{bd}}\right)^{\beta},$$

with t_{REM} indicating the amount of sleeping time spent in the REM phase, and, conversely, $t_{non-REM}$ indicating the amount of sleeping time not spent in that phase. The latter quantity is easily computed by $t_{tot,i} - t_{REM,i}$ for $i = 1, \ldots, n$, where n indicates the sample size of 41

mammalian species. The choice of taking this type of ratio was driven by the analyses of Savage and West [2007], referring to how they handled the sleep awake time ratio.

A possible explanation for this link could reside into the asymmetry between body and brain's metabolic rates. Even though larger body masses are correlated with less sleeping time, higher brain metabolic rates might request a higher portion of REM sleep in order to operate the necessary amount of reorganization of the brain damage reparation. Furthermore, since during REM sleep brain's metabolic activities are on, it might also be that mammals which are characterized with a low brain body mass ratio simply have a more active brain during sleep. An example in that regard is the Asian Elephant, which is one of the mammals which spend the least time asleep, roughly 4 hours, but has to devote a large portion of its sleep to the REM phase, having a small brain body mass ratio.

Since the main relationship is a power law, it is hence possible to build a linear model by taking the logarithm of the involved quantities, with the exponent of that relationship, β, now becoming a regression coefficient. Also the lifespan and gestation covariates were taken in logarithm to reduce their range, all covariates were also centred, before applying any method. The full model we started the analysis from is then, with the addition of an intercept:

$$\log\left(\frac{t_{REM}}{t_{non-REM}}\right) = \beta_0 + \beta_1 \log\left(\frac{M_{br}}{M_{bd}}\right) + \beta_2 \log L + \beta_3 \log G + \beta_4 P + \beta_5 E + \beta_6 D + \epsilon_i \quad \text{for} \quad i = 1, \ldots, 41.$$

The estimated coefficients of the ordinary least squares of the full model are reported in Table 4.5. Only the coefficient associated to brain body mass ratio is considered statistically significant, at a 5% confidence level, alongside the intercept term. It is also the coefficient that has a low estimated standard error, at 0.083, and hence its confidence interval does not contain the 0. Even though the theory suggests that also the danger index should be taken into account, its associated coefficient has a very high standard deviation, at 0.218, and its coefficient interval crosses the 0. The residuals show the presence of homoschedasticity, and their estimated standard deviation is set to 0.447. Moreover, the R^2 of the full model is fairly low, at 30%, and the zero slopes hypothesis is not rejected. The mean value of our response variable is set at -1.61, hence the actual ratio is equal to 0.20, meaning that the amount of sleep spent in the REM phase is expected to cover one sixth of the total sleeping time; to this regard, man is above the average, with a ratio of 0.31, hence with the REM phase occupying slightly less than one fourth of the total sleeping time. These results prove the importance of the mammals' metabolic rates in dictating their amount of REM sleep,

	OLS	OLS_{AFIC}	AIC	Lasso	Ridges
Brain body mass ratio	-0.298^*	-0.293	-0.307	-0.266	-0.288
Lifespan	0.063				
Gestation	-0.160	-0.143		-0.05	-0.135
Predation	0.047				
Exposure	0.087	0.109			0.098
Danger	-0.282	-0.251	-0.251	-0.151	-0.239

Table 4.5: Estimates of the coefficients associated with the selected variables by all methods. The absence of the value means that the covariate is not present in the method's final model. The two stars on the brain body mass ratio coefficient in the OLS column indicate that only that covariate was deemed statistically significant, at a 5% confidence level.

as they are the only covariate deemed to have a strong influence on the outcome from the full model. The theory, however, suggests that also other traits could impact on the sleeping behaviour of mammals, hence covariate selection is needed to check if some of these other covariates could be identified.

The first model selection method used is an application of our AFIC approach on the ordinary least squares. In order to adapt the method to the ordinary least squares estimator, the value of the tuning parameter, λ, is forced to be zero. Then we proceeded to use the AIC criterion and the lasso estimator. Finally our ridges method was used, with an ordinary least squares plug-in and minimizing the expected estimation error, as the simulations showed no particular differences between the choices of plug-in and criterion in the low dimensional case. As we presented in Section 3.1.1, we expect the AIC to select a parsimonious model whereas the lasso is expected to maybe include more covariates, in comparison. Eventually, since the simulations showed that the ridges has a tendency to be generous with the sizes of its selected model, we expected our AFIC based methods to carry over this property.

The AIC method selects only two covariates which are the brain body mass ratio and the danger index, accordingly with our expectations. Including also the danger effect in the final model, the AIC is able to attribute the brain body mass ratio a larger coefficient (in terms of absolute value), meaning that the danger status of a mammal has to be taken into account to explain its sleeping behaviour. The lasso, however, introduces one more covariate, gestation, with a negative coefficient. This might be in line with the fact that longer gestation periods are also associated with the developing of a more physically structured offspring, hence

having slower metabolic rates and needing less REM activity during their sleep. Siegel [2005] suggests that a particular subgroup of mammals which has longer gestation periods, namely the *altricial* mammals, have more REM activity during the day than other mammals. Since our target variable refers to the quantity of REM activity during sleep, we expect the larger amount of REM activity of the altricial mammals to not cover a large part of their sleeping time. The ordinary least squares model found by the AFIC, however, includes another covariate to its chosen subset, which is the exposure index. The positive coefficient estimated for that covariate is certainly against the expectations. As we noted earlier, mammals that are more exposed to danger during sleep limit their REM activities, this contrasts could be resolved in favour of a more prominent role of the overall danger a mammal experiences rather than the danger it is subject to solely during sleep.

Then our ridges method was applied, with the subset that minimized the criterion being found through exhaustive search. Judging by Table 4.5, our method selects the least parsimonious model of the analysis together with the other AFIC method, connected to ordinary least squares. Both AFIC method introduce the exposure index into their selections. Moreover, the similarity of the coefficients' estimates show that was introduced a small amount of shrinkage. The ridges, as expected, reaches consistently lower values of the AFIC criterion as Figure 4.2a and Figure 4.2b show, according to the theory we presented in Chapter 2. Those AFIC plots present the two estimates of the two most important covariates, the brain body mass ratio and the danger index, plotted against the square roots of the AFIC scores associated to every subset. For the sake of a more plain visualization, the subsets that did not contain the focus covariates were removed from each plot. Note that when the subsets contain few variables, with the narrow models being the extreme case, the scores of the ridges and the ordinary least squares are approximately the same, whereas when the subsets sizes start to increase the two methods diverge. The effect of the shrinkage is particularly apparent in the improvement of the AFIC score associated with the full model. When the focus is the brain body mass ratio, it has roughly the same score as the narrow model, while it is slightly preferred to the narrow model containing the danger index. When using the ridges, both full models are consistently better than both narrow models. Furthermore, both the full and the narrow models have an associated score that it is rather distant from the best model, and this setting leads to the conclusion that an AFIC based method can be utilized to perform a meaningful variable selection. This property also came out in the simulation studies, where we proved that this method is capable of including the real active set in the final chosen model.

In order to see how the AFIC approach of the ridges procedure compares with the other model selection methods it is possible to turn to the AFIC plots shown in Figure 4.2c and in Figure 4.2d. This time, only the ridges scores are reported, and the models selected by AIC and lasso procedure are highlighted. It is seen that both selections are very close to the ridges optimal model, with the lasso selecting the second ranked model and the AIC the fifth one. Again, all selections are distant from the score of the full model. This outcome is encouraging, as our method does not take a completely different direction from established model selection methods, but offers a different view on the problem.

Lastly, an interesting comparison that can be made refers to the difference of the estimates of the regression coefficients performed by the AIC and lasso, and the ridges built on their selected subsets. The ridges regression that uses the AIC model produces roughly the same estimates of the to focussed coefficients. This is certainly due to the scarcity of covariates present in that subset, hence shrinkage is not needed to be large. The ridges regression computed on the lasso selected subset, on the other hand has give very different estimates. This is caused by the different role the tuning parameter has in the two methods, and by how it is selected. In order to fine-tune the lasso, leave-one-out cross-validation was used, hence its estimates should be regarded when the final aim is prediction. Since the ridges is fine-tuned minimizing the expected estimation error, its estimates should be less biased. In this way, the ridges method could be seen as a link between penalized an non penalized regression: it is always a better choice than ordinary least squares since we showed that improves the value that the criterion assumes, but it is also able to give stable estimates without introducing unnecessary amounts of shrinkage.

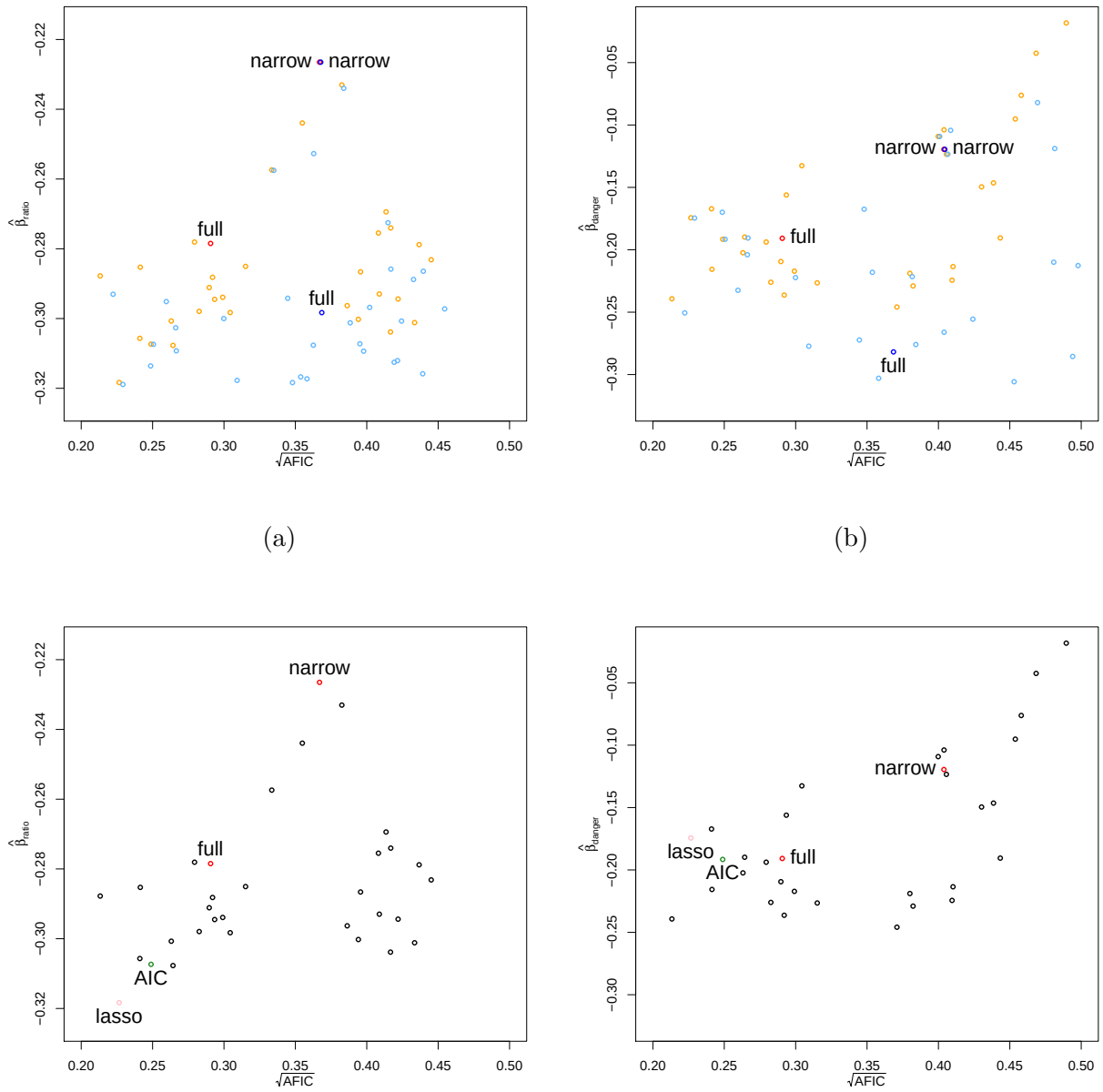


Figure 4.2: (a) OLS (light blue) and ridges (orange) AFIC plots with the AFIC estimate of brain body mass ratio coefficient on the y axis. The full model containing all the covariates and the narrow model containing only this covariate are also indicated for both methods. (b) OLS and ridges AFIC plots with AFIC estimate of danger coefficient on the y axis. Full and narrow models are highlighted. (c) Ridges AFIC plot with same structure as in (a). The coloured dots indicate the subset selected by AIC and lasso. (d) Ridges AFIC plot with same structure as in (c). AIC and lasso selections are highlighted.

Chapter 5

Concluding remarks

In this thesis, we have presented the mean squared error minimization method in order to fine-tune the penalty parameter, λ, of ridge regression, while in Chapter 3, we suggested that this method could also be interpreted as an averaged focussed criterion which can be used to perform model selection.

We proposed that minimizing the mean squared error of the ridge estimates could be a viable alternative to other methods in order to fine-tune the penalty parameter, λ. Through simulations we assessed that the oracle tuning parameter yields the minimum value of the mean squared error, in regards to the criteria used. When the target is a lower estimation error, expected estimation error must be used, conversely, when the target is prediction, expected prediction error would be the proper choice as criterion to minimize. This technique seemed to be competitive also for predicting hold-out-data. In this case the oracle tuning shows to yield a lower hold-out prediction error than cross-validation, which was not expected. Another interesting result referred to the apparent best out-of-sample predictive performance of the expected estimation error: its oracle tuning has a lower prediction error than the oracle tuning of the expected estimation error, but when plug-in estimates are used the latter criterion is to be preferred to the former.

This criteria can also be used as an averaged focussed criteria to perform model selection. To this regard, the criteria have to be minimized jointly over the possible subsets one wants to explore and over the penalty parameter, λ. In this case the subset containing the active variables, or the most important variables should be selected by the procedure as it is expected to yield the lowest mean squared error. Also in this case, simulations demonstrated that this could be a viable path to perform model selection, as the ridge regression framework supplies the possibility of being used also in situations in which the correlation between

the covariates is high and in high dimension. Since in this case computational issues arise we proposed reduced forms to compute the mean squared error based criteria. With these formulations, it is possible to sensibly reduce the computational times of the procedure and hence practical applications are more feasible. Speaking of its model selection performance, we found that every combination of criteria and plug-in used is very sensitive to the signal strength of the regression parameter. To this point, the true variables selection rates are stable but our methods fail to recognize every active variable when the signal is fairly low. Eventually, in this framework using a stepwise algorithm seems to be a good choice, as yields a performance comparable with exhaustive searches, and with the use of the mean squared errors in their reduced form, its computational times required are in the order of the seconds, even when p is relatively large.

Eventually, it is possible to produce a *focussed information criterion* when referring to subset selection performed through the focussed fine-tuning criterion of the fridge [Hellton and Hjort, 2018]. This setting is promising, as it could potentially perform *personalised model selection*, tailoring what covariates should be more important for a particular new observation.

The main fields that are left to explored are the use of other criteria to minimize, both for fine-tuning and as AFIC criteria, better techniques to estimate the squared bias term of these criteria, and as we suggested in Chapter 4, better techniques to explore the possible models that are generated from p covariates.

Further developments

As reported by Claeskens and Hjort [2008] when presenting the FIC, every direct estimate of the squared bias term of any mean squared error formulation suffers from the problem:

$$\mathbb{E}[\widehat{\text{Bias}^2}] = \text{Bias}^2 + \mathbb{V}[\widehat{\text{Bias}}].$$

Since all of the criteria we used are different formulations of mean squared errors, corrections for the estimate of their bias squared term ought to be explored.

Merely subtracting the variance of the estimate from the estimated term could produce negative squared bias estimates, hence a first correction that might be applied involves the subtraction of the variance term and the truncation at zero:

$$\widehat{\text{Bias}^2}_+ = \max\{\widehat{\text{Bias}^2} - \mathbb{V}[\widehat{\text{Bias}}], 0\}.$$

This type of correction, however, may cause problems during the practical implementations. Since the criteria are minimized through the use of optimization routines, those algorithms might not work properly if the truncation of the squared bias term creates discontinuities in the function, of the estimated mean squared errors. For this reason, as suggested by Hellton and Hjort [2018], a more complex smooth correction can be used, such as:

$$\widehat{\text{Bias}^2}_+ = \widehat{\text{Bias}^2} - \mathbb{V}[\widehat{\text{Bias}}]\frac{\widehat{\text{Bias}^2}}{\widehat{\text{Bias}^2} + 1}.$$

Both corrections, the latter in particular, introduce more terms to compute into the criteria formulations. This necessarily causes a significant increase in computational time required, especially if one wants to perform model selection, where, as we already stated, several minimizations are needed. Even though the estimation of the squared biased term never constituted a problem during the simulations and applications presented in this work, it is certainly a problem that needs to be solved. The solution might be found in exploring new ways to reduce the computational costs of the minimization procedure, in the form of new decompositions to apply to the elements involved into the estimation of the mean squared error, as alternatives to the singular value decomposition for the design matrix $\mathbf{X}$.

As we saw in Chapter 4, finding the right algorithm to explore the subset space could be a very difficult task as the number of available covariates, p, increases. During the simulations, we used our methods assuming to have a low amount of a priori information. In some situations, however, it could be needed to have guidance in order to perform a fast search through all the possible subsets.

Since we already stated that a product of using exhaustive search or forward algorithms might be the inclusion of noise variables, a first solution is to limit the size of the subsets explored. When no a priori information about the amount of true variables is given, univariate regressions might be taken into account as a way to explore which, and how many, covariates carry a signal. This task can be accomplished without leaving the AFIC framework, as it is possible to compute which univariate regressions have the lowest AFIC score. Another possible application of the use of univariate regressions might be to rank the variables, referring to their AFIC scores, then creating a *protected set* containing an arbitrary number of high ranked covariates, which is included in every subset that method searches among.

As a final suggestion, another possible path worth exploring consists of combining the random subset search with forward or backward algorithms. Stepwise algorithms might not be the most accurate in identifying the true model but are nonetheless computationally

feasible in most occasions. By starting with several randomly generated subsets it might be interesting to compare if both forward and backward algorithms converge to select the same variables, or even the same amount of variables. In situations when p has a really large value (typically when $p > 10$), the process could still be implemented by stopping the stepwise algorithms after a controlled number of iterations, or replicating it in several iterations, each applied to a randomly chosen subset containing a smaller number of covariates.

The mean squared error minimization framework we presented, can be extended to other forms of penalized regression techniques such as lasso and logistic ridge regression. These latter two methods, however, do not possess any enclosed form solution of their optimization problems, hence they do not have a definition of their estimator of the regression parameter $\boldsymbol{\beta}$. This problem may limit their use as in those cases it is not possible to have a precise expression for the mean squared error to minimize. In order to apply our methods, however, it is possible to estimate the mean squared error functions via bootstrap replications. As Hellton and Hjort [2018] suggest, one could use the plug-in estimates $\tilde{\boldsymbol{\beta}}$ to produce new sets of $\boldsymbol{y}$ trough parametric bootstrap, hence resampling the residuals obtained by subtracting $\mathbf{X}\tilde{\boldsymbol{\beta}}$ from the actual response variable values $\boldsymbol{y}$ and then compute, through several iteration, the mean squared error of the estimates obtained on the bootstrap generated sample.

This way it would be possible to extend the ridges methodology also in a logistic ridge regression framework, but further research is needed. To this regard, computational issues were a problem also when bootstrap replications were not needed, hence in this situation the application of the method is currently infeasible. However, it is indeed a path worth exploring as we found that it could produce interesting results.

Bibliography

H. Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.

H. Akaike. On entropy maximization principle. *Applications of statistics*, pages 27–41, 1977.

H. Akaike. A new look at the bayes procedure. *Biometrika*, 65(1):53–59, 1978.

T. Allison and D. V. Cicchetti. Sleep in mammals: ecological and constitutional correlates. *Science*, 194(4266):732–734, 1976.

G. Claeskens and N. L. Hjort. *Model selection and model averaging*, volume 330. Cambridge University Press Cambridge, 2008.

R. Farebrother. Further results on the mean square error of ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(3):248–250, 1976.

K. H. Hellton and N. L. Hjort. Fridge: Focused fine-tuning of ridge regression for personalized predictions. *Statistics in medicine*, 2018.

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

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

V. M. Savage and G. B. West. A quantitative, theoretical framework for understanding mammalian sleep. *Proceedings of the National Academy of Sciences*, 104(3):1051–1056, 2007.

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

J. M. Siegel. Clues to the functions of mammalian sleep. *Nature*, 437(7063):1264, 2005.

C. Theobald. Generalizations of mean square error applied to ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 103–106, 1974.

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

W. N. van Wieringen. Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169*, 2015.