FINAL DEGREE PROJECT

Sparse Methods in Classification and Regression

Presented by:
Antonio Rivero Martínez

Supervised by:
DR. EMILIO CARRIZOSA PRIEGO

FACULTY OF MATHEMATICS
Statistics and Operational Research Department
Sevilla, June 2021

Contents

Abstract **5**

Introduction **7**

1 The Linear model **9**
1.1 Logit Model 11

2 Basic regularization methods **13**
2.1 Ridge Regression 13
2.2 The Lasso 14
2.2.1 The lasso in the logit model 15
2.3 The Group Lasso 16

3 Naive elastic net **17**
3.1 Definition 17
3.2 Solution 18
3.3 The grouping effect 19

4 Elastic Net **23**
4.1 Deficiency of the naive elastic net 23
4.2 The elastic net estimate 23
4.3 Elastic net in logit model 26
4.4 Comparison between Lasso and Elastic net 26
4.4.1 Introduction and Notation 26
4.4.2 Comparison 27

5 Elastic net in R studio **29**
5.1 Elasticnet package 30
5.2 glmnet package 34

6 Numerical examples **35**
6.1 Prostate Database 35
6.1.1 Description 35

6.1.2 Ordinary Least Square regression 36
6.1.3 Lasso regression model 39
6.1.4 Elastic Net Regression 42
6.2 Hitters database 43
6.2.1 Description 43
6.2.2 Ordinary Least Square regression 43
6.2.3 Lasso regression model 48
6.2.4 Elastic Net Regression 50
6.3 Simulation study 51
6.3.1 Ordinary Least Square regression 51
6.3.2 Lasso regression model 53
6.3.3 Elastic Net Regression 56

Bibliography **59**

Abstract

The regression problem with a large number of variables appears in various fields of science, sparse methods make this problem more interpretable and more precise. In this work we present the method $Elastic\ Net$, which outperforms the Lasso in some situations. The elastic net have the grouping effect, while lasso does not, this is that strongly correlated predictors tend to "behave" in the same way. The lasso does not work well when the number of predictors is much grater than the number of observations, $p \gg n$. However, elastic net is useful in this situation.

Introduction

The sentence "I've got all these variables, but I don't know which ones to use", and the question "How can one improve the performance of the model available" could define the basis of this work.

Therefore, the goal is to find a model that helps to make these decisions, that is, to decide which variables are significant and which ones are not. The regression problem with a large number of variables appears in various fields of science. This phenomenon is occurring more and more frequently due to advances in technology.

Some of the requirements in a variable selection model are:

- More interpretable models
- More accurate predictions
- Stability, in the sense small changes in the data should not lead big changes in the predictors.

Traditional variable selection methods, such as ridge regression, fail in one or more of the above requirements. Modern procedures such as Lasso (Tibshirani, 1996), generally improve stability and predictions. Group Lasso is a natural extension of Lasso, which selects the variables in a grouped way. In section 2 the methods named on this paragraph will be presented.

Although Lasso works successfully in many occasions, it has some limitations:

- The number of predictors may be much grater than the number of observations ($p \gg n$)
- Explanatory covariates may be strongly correlated.

Therefore, between section 3 and section 4, it is presented the model Elastic Net in order to overcome these limitations.

In section 5 there is an explanation of the R-studio package $elasticnet$. Finally, to fix ideas, in section 6 there are two numerical examples using popular data sets from the literature and a simulation case.

Chapter 1

The Linear model

Consider the linear model, which is expressed as:

$$y = \alpha + \mathbf{x}^t\beta + \epsilon \tag{1.1}$$

where:

- $\mathbf{x}$ is the predictor vector
- β is a p-dimensional unknown parameter
- $\alpha \in \mathbb{R}$ is unknown
- ϵ is the error vector, which its mean is 0 and its variance is σ^2

From a group of pairs (x_1, y_1),. . ., (x_n, y_n), the ordinary least squares (OLS) consist on finding α and β which minimizes the sum of the square of the errors between the data $\mathbf{y}_j$ and the predictions $\hat{\mathbf{y}}_j = \alpha + \mathbf{x}_j^t\beta$.
The OLS estimator $\hat{\beta}$ is obtained by solving the next optimization problem

$$\underset{\alpha\in\mathbb{R},\ \beta\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{n}(y_i - \alpha - \mathbf{x}_i^t\beta)^2 \tag{1.2}$$

Let $\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & \mathbf{x}_1 \\ \vdots & \vdots \\ 1 & \mathbf{x}_n \end{bmatrix}$$

Now, the problem (1.2) is:

$$\underset{\alpha\in\mathbb{R},\beta\in\mathbb{R}^p}{\arg\min} \left(\mathbf{y} - \tilde{\mathbf{X}}^t\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\right)^t \left(\mathbf{y} - \tilde{\mathbf{X}}^t\begin{pmatrix}\alpha\\ \beta\end{pmatrix}\right) \tag{1.3}$$

Necessary and sufficient condition of optimality

Calling $f(\alpha,\beta)$ the objetive function in (1.3), it is a convex and differentiable function. Therefore, a necessary and sufficient condition for (α,β) to be an optimal solution of (1.2) is:

$$\nabla f(\alpha, \beta) = 0 \tag{1.4}$$

Using the hypothesis that $\mathbf{X}^t\mathbf{X}$ is invertible, (1.4) has an unique solution.

$$\begin{pmatrix} \hat{\alpha} \\ \beta \end{pmatrix} = (\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^t\mathbf{y} \tag{1.5}$$

In the following lines, (1.5) is proved.

$$f(\alpha, \beta) = \left(\mathbf{y} - \tilde{\mathbf{X}}^t\begin{pmatrix} \alpha \\ \beta \end{pmatrix}\right)^t \left(\mathbf{y} - \tilde{\mathbf{X}}^t\begin{pmatrix} \alpha \\ \beta \end{pmatrix}\right)$$

Denoting $\binom{\alpha}{\beta}$ as θ:

$$f(\theta) = \left(\mathbf{y} - \tilde{\mathbf{X}}^t\theta\right)^t \left(\mathbf{y} - \tilde{\mathbf{X}}^t\theta\right) = \sum_{i=1}^{n}(\mathbf{y}_i - \tilde{\mathbf{x}}_i^t\theta)^2$$

So;

$$\frac{\delta}{\delta\theta_k}f(\theta) = 2\sum_{i=1}^{n}(\mathbf{y}_i - \tilde{\mathbf{x}}_i^t\theta)(-\tilde{\mathbf{x}}_i k)$$

Therefore;

$$\nabla f(\theta) = -2\tilde{\mathbf{X}}^t(\mathbf{y} - \tilde{\mathbf{X}}\theta)$$

$$\nabla f(\theta) = 0 \Leftrightarrow -2\tilde{\mathbf{X}}^t(\mathbf{y}-\tilde{\mathbf{X}}\theta) = 0 \Leftrightarrow \tilde{\mathbf{X}}^t(\mathbf{y}-\tilde{\mathbf{X}}\theta) = 0 \Leftrightarrow \tilde{\mathbf{X}}^t\mathbf{y} = \tilde{\mathbf{X}}^t\tilde{\mathbf{X}}\theta \Leftrightarrow \theta = (\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^t\mathbf{y}$$

1.1 Logit Model

A Logit (logistic) model is useful to examine how prediction variables influence a binary response $\mathbf{y}$. This response may take the values 1 and 0 to denote the existence or a lack of a certain qualitative characteristic (a woman can be pregnant or not, ...). The logit model is created to estimate the probability of $\mathbf{y}$ = 1 with a logistic function of linear combinations of $\mathbf{x}$.

Consider a binary response variable $\mathbf{y}$ (n$\times$1) (the value $\mathbf{y}$ = 1 indicating the existence of a qualitative characteristic and the value $\mathbf{y}$ = 0 indicating the lack of it). This model assumes that the probability for observing y_i = 1, with $\mathbf{x}_i$= $(x_{i1}, \ldots, x_{ip})^T$, is given by a logistic function of a linear combination of $\mathbf{x}$.

$$p(\mathbf{x}_i) = P(y_i = 1|\mathbf{x}_i) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})} \tag{1.6}$$

The next function implies the probability of the lack of the characteristic:

$$1 - p(\mathbf{x}_i) = P(y_i = 0|\mathbf{x}_i) = \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})} \tag{1.7}$$

With (1.6) and (1.7):

$$\log\left\{\frac{p(\mathbf{x}_i)}{1 - p(\mathbf{x}_i)}\right\} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \tag{1.8}$$

(1.8) indicates that the logit model is equivalent to a log-linear model for the odd ratio $p(\mathbf{x}_i)/(1 - p(\mathbf{x}_i))$. A positive value of β_j indicates that the variable $\mathbf{x}_j$ will assist the existence of the characteristic as the odd ratio increase. A β_j = 0 corresponds to the lack of an effect of the variable $\mathbf{x}_j$ on the qualitative characteristic. For independent and identically distributed observations, the probability function is:

$$L(\beta_0, \beta) = \prod_{i=1}^{n} p(\mathbf{x}_i)^{y_i}(1 - p(\mathbf{x}_i))^{1-y_i}$$

The maximum probability estimators (β) are obtained from solving the maximization problem $(\hat{\beta}_0, \hat{\beta}) = \arg\max_{\beta_0,\beta} \log L(\beta_0, \beta)$ where:

$$\log L(\beta_0, \beta) = \sum_{i=1}^{n} \left[y_i \log\{p(\mathbf{x}_i)\} + (1 - y_i) \log\{1 - p(\mathbf{x}_i)\}\right] \tag{1.9}$$

Chapter 2

Basic regularization methods

This chapter will present three basic shrinkage methods, Ridge Regression, the Lasso and the Group Lasso. These methods are used since retaining a subset of the predictors and discarding the rest makes a more interpretable model and it probably has lower prediction error than the full model. Furthermore, those methods are more continuous than subset selection and they are not affected by high variability.

2.1 Ridge Regression

This method was the first regularization method introduced in statistics [4]. In Ridge regression the coefficients are shrunk by imposing a penalty on their size. Those coefficients minimize a sum of square in which a penalization term is added.

$$\hat{\beta}^{ridge} = \arg\min_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\} \tag{2.1}$$

In this equation, $\lambda \geq 0$ is a parameter which manages the amount of shrinkage, as λ rises, the amount of shrinkage increases. It is equivalent to write the ridge problem as:

$$\hat{\beta}^{ridge} = \arg\min_{\beta}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2, \text{ subject to } \sum_{j=1}^{p}\beta_j^2 \leq s, \tag{2.2}$$

There is a one to one correspondence between λ in (2.1) and s in (2.2). When there are many correlated variables, their coefficients can exhibit high variance. The solutions of this problem is not equivariant under scaling of the predictors. As a result, we can standardize the predictors before solving (2.1). In addition, observe that β_0 is not in the penalty term, so we can use centered predictors, each x_{ij} gets replaced by $x_{ij} - \bar{x}$. and β_0 is estimated by $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. The other coefficients are estimated by a ridge regression, using the centered x_{ij}. As the centering has been done, the matrix $\mathbf{X}$ has p

columns, because the first column is 0 after the centering.

Writing (2.1) in matrix form,

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta \tag{2.3}$$

The solutions are:

$$\hat{\beta}^{ridge} = \mathbf{R}\mathbf{y} \tag{2.4}$$

with

$$\mathbf{R} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$$

where $\mathbf{I}$ is the p×p identity matrix. Observe that with the choice of the penalty term ($\beta^T\beta$), the solution is a linear function of $\mathbf{y}$. Ridge regression makes the problem nonsingular, even if $\mathbf{X}^T\mathbf{X}$ is a singular matrix, as the solution adds a constant to the diagonal of $\mathbf{X}^T\mathbf{X}$ before the inversion.

The problem of that method is that the coefficients are shrunk toward zero when $\lambda \to \infty$, but they would not be exactly zero.

2.2 The Lasso

The Lasso is a shrinkage method as ridge regression, with slight but important differences. The Lasso estimated is:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2, \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq s, \tag{2.5}$$

As in ridge regression, the coefficients can be standardized in order to re-parametrize β_0, which is $\bar{y}$. The Lasso problem can be written in the *Lagranganian form*:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p}|\beta_j| \right\}. \tag{2.6}$$

Observe the similarity to the ridge regression problems (2.1) and (2.2). While in Ridge regression the penalty is $\sum_{j=1}^{p}\beta_j^2$, in Lasso is $\sum_{j=1}^{p}|\beta_j|$. That difference causes a nonlinear solution in the y_i. If we make s sufficiently small in (2.5) some coefficients will be exactly zero. Therefore, the Lasso is like a continuous subset selection. If s is chosen bigger than $s_0 = \sum_{j=1}^{p}|\hat{\beta}_j|$, where $\hat{\beta}_j$ is the solution of (OLS), the Lasso solution are those $\hat{\beta}_j$. However, if s = $\frac{s_0}{2}$, then the coefficients are shrunk by about 50% on average [11]. Lasso moves each coefficient by a constant λ, truncating at zero. It is good as if a coefficient is zero the problem become easier.

2.2.1 The lasso in the logit model

The lasso can be extended to the logit model. The linear predictor $\mathbf{X}\beta$ is related to the conditional mean μ of the variable $\mathbf{y}$ through the logit function $\log(\mu/(1-\mu))$. Since $\mathbf{y}$ is a binary variable, it is binomial-distributed and $\mu = p(\mathbf{x}_i)$. Therefore, the logit model for $\mathbf{y}$ is the same as it is defined in (1.8)

The lasso estimate for the logistic model is obtained by solving this optimization problem:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} g(-\mathbf{y}_i \mathbf{x}_i^T \beta) \right\}, \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq s, \tag{2.7}$$

where s $\geq$ 0 is the tuning parameter, and $g(u)$ = log(1 + exp(u)) is the log-loss function. An equivalent representation of the lasso estimate $\hat{\beta}$ in the logit model is:

$$\arg\min_{\beta} \left\{ \sum_{i=1}^{n} g(-\mathbf{y}_i \mathbf{x}_i^T \beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.8}$$

In 2003 an asymptotically convergent algorithm to solve the optimization problem (2.8) was developed. The details can be found in [5]

Another way to obtain the lasso estimate in the logit model is by maximizing the probabilistic function of the logit model (1.9) with lasso constraint.

Let $l(\beta)$ = log$L(\beta)$. The Lasso estimate, β, is obtained by maximizing the penalized log-probabistic function:

$$\hat{\beta} = \arg\max_{\beta} \left\{ \sum_{i=1}^{n} l(\beta) \right\}, \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq s, \tag{2.9}$$

It can be solved by a nonlinear programming procedure. An equivalent representation of the lasso estimate $\hat{\beta}$ in the logit model is:

$$\arg\max_{\beta} \left\{ \sum_{i=1}^{n} l(\beta) - \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.10}$$

2.3 The Group Lasso

Group Lasso is motivated by the fact that predictors can occur in some groups, so it would be better to obtain a solution which uses only a few of the groups. Assume that there are K groups and the coefficients vector is structured as:

$$\beta^G = (\beta_1^T, \ldots, \beta_K^T)^T$$

A sparse set of groups is produced, and all entries of β_k, $k = 1, \ldots, K$ are nonzero or all of them are zero. The group Lasso problem is formulated as:

$$\min_{\beta} \left\| \mathbf{y} - \sum_{k=1}^{K} \mathbf{X}^{(k)} \beta^{(k)} \right\|_2^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k} \|\beta^{(k)}\|_2 \tag{2.11}$$

where $\mathbf{X}^{(l)}$ is the submatrix of $\mathbf{X}$ with the columns corresponding to the variables in group l, $\beta^{(l)}$ the coefficient of group k and p_k is the length of $\beta^{(k)}$. In this criterion, the objetive function is nonsmooth for $\beta^{(k)}$ = 0 in $\|\beta^{(k)}\|_2$. The sparsity is determined by λ. If p_k = 1 for each group, this criterion gives the lasso solution.

The computation of the solution involves calculating the necessary and sufficient KKT conditions for $\hat{\beta}$ to be a solution of (2.11). The solution of the KKT condition is:

$$\hat{\beta}_k = \left(\frac{\lambda \sqrt{p_k}}{\|\hat{\beta}_k\|} + (\mathbf{X}^{(k)})^T \mathbf{X}^{(k)} \right)^{-1} (\mathbf{X}^{(k)})^T \hat{r}_k \tag{2.12}$$

where $\hat{r}_k$ is defined as $\hat{r}_k$ = **y** - $\sum_{l \neq k} \mathbf{X}^{(k)} \hat{\beta}_k$. To obtain a full solution, it was proposed to use a blockwise coordinate descent algorithm which applies the estimate (2.12) to k = 1 , . . . , K

Chapter 3

Naive elastic net

3.1 Definition

Let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the response and $\mathbf{X} = [\mathbf{x}_1 | \ldots | \mathbf{x}_p]$ a matrix, in which $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$, $j = 1, \ldots, p$ are the predictors. One can standardize $\mathbf{y}$ and $\mathbf{X}$, so it is assumed that the response is centered and the predictors are standardized,

$$\sum_{i=1}^{n} y_i = 0, \ \sum_{i=1}^{n} x_{ij} = 0, \ \text{and} \sum_{i=1}^{n} x_{ij}^2 = 1, \ \text{for } j = 1, 2, \ldots, p. \tag{3.1}$$

For any fixed $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, the naive elastic net criterion is defined as:

$$L(\lambda_1, \lambda_2, \beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1 \tag{3.2}$$

where

$$\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2 \ \text{ and } \ \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$$

The naive elastic net estimator $\hat{\beta}$ is the minimizer of (3.2).

Let $\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}$. Solving $\hat{\beta}$ in(3.2) is the same as optimizing:

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2, \ \text{ subject to } \ \alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1 \leq s \ \text{ for some s.} \tag{3.3}$$

The function $\alpha\|\beta\|_2^2 + (1-\alpha)\|\beta\|_1$ is called elastic net penalty, which is a convex combination of ridge and Lasso penalty. When $\alpha = 1$ naive elastic net becomes ridge regression, and when $\alpha = 0$ naive elastic net becomes the Lasso. For $\alpha \in [0,1)$, the elastic net penalty function does not have first derivate at 0 and it is strictly convex for $\alpha \geq 0$

3.2 Solution

This section shows a method to solve the naive elastic net problem. It becomes the solution equivalent to a lasso type, so the naive elastic net solution has also computational advantages as lasso.

Lemma 3.2.1. *Let* $(\mathbf{y}, \mathbf{X})$ *and* (λ_1, λ_2) *be given data, define an artificial data* $(\mathbf{y}^*, \mathbf{X}^*)$ *by*

$$\mathbf{X}^*_{(n+p)\times p} = \frac{1}{1+\lambda_2}\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}, \quad \mathbf{y}^*_{(n+p)} = \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix}$$

Let $\gamma = \frac{\lambda_1}{\sqrt{1+\lambda_2}}$ *and* $\beta^* = \sqrt{1+\lambda_2}\beta$*, then the naive elastic net can be written as*

$$L(\gamma, \beta) = L(\gamma, \beta^*) = \|\mathbf{y}^* - \mathbf{X}^*\beta^*\|^2 + \gamma\|\beta^*\|_1 \tag{3.4}$$

Let

$$\hat{\beta}^* = \arg\min_{\beta^*} L(\gamma, \beta^*)$$

then

$$\hat{\beta} = \frac{1}{\sqrt{1+\lambda_2}}\hat{\beta}^*$$

Proof

In this proof one starts in equation (3.4), and one has to finish in equation (3.2).

$$\|\mathbf{y}^*-\mathbf{X}^*\beta^*\|^2+\gamma\|\beta^*\|_1 = \left\|\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} - (1+\lambda_2)^{-\frac{1}{2}}\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}\sqrt{(1+\lambda_2)}\beta\right\|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\left\|\sqrt{(1+\lambda_2)}\beta\right\|_1$$

Simplifying:

$$\|\mathbf{y}^* - \mathbf{X}^*\beta^*\|^2 + \gamma\|\beta^*\|_1 = \left\|\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}\beta\right\|^2 + \lambda_1\|\beta\|_1$$

Working on the first addend:

$$\left\|\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}\beta\right\|^2 = \left[\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}\beta\right]^t\left[\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} - \begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}\end{pmatrix}\beta\right] =$$

$$(\mathbf{y} - \mathbf{X}\beta)^t(\mathbf{y} - \mathbf{X}\beta) + (0 - \sqrt{\lambda_2}\mathbf{I}\beta)^t(0 - \sqrt{\lambda_2}\mathbf{I}\beta) =$$

$$(\mathbf{y} - \mathbf{X}\beta)^t(\mathbf{y} - \mathbf{X}\beta) + (0 - \sqrt{\lambda_2}\beta^t\mathbf{I})(0 - \sqrt{\lambda_2}\mathbf{I}\beta) =$$

$$\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2\beta^t\beta = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2\|\beta\|^2$$

Finally replacing , (3.2) is obtained □

Lemma 3.2.1 states that naive elastic net problem can be transformed to an equivalent lasso problem. The sample size in the expanded problem is n+p and $\mathbf{X}^*$ has rank p, this means that naive elastic net can select all p predictors in all situations. This Lemma also shows that naive elastic net can do an automatic variable selection similar to lasso. In the next section it is shown that the naive elastic net has the "grouping" effect, which means that it select correlated variables, this property is not shared by lasso.
In the case of an ortogonal design, the naive elastic net solution is:

$$\hat{\beta}_i(\text{naive elastic net}) = \frac{(|\hat{\beta}_i(ols)| - \frac{\lambda_1}{2})_+}{1+\lambda_2}\text{sgn}(\hat{\beta}_i(ols)) \tag{3.5}$$

where $\hat{\beta}_i(ols) = \mathbf{X}^T\mathbf{y}$ and $()_+$ denotes the positive part. The solution of ridge regression is given by $\hat{\beta}(\text{ridge}) = \frac{\hat{\beta}(ols)}{(1+\lambda_2)}$, and the lasso is $\hat{\beta}_i(\text{lasso}) = (|\hat{\beta}_i(ols)| - \frac{\lambda_1}{2})_+\text{sgn}(\hat{\beta}_i(ols))$.

3.3 The grouping effect

In the "large p, small n" problem, the grouped variables situation is an important corcern, which has appeared a number of times in the literature. In this section we consider the generic penalization method.

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda J(\beta) \tag{3.6}$$

where J(·) is a functional and $\lambda \geq 0$

A regression method presents the grouping effect if the regression coefficients of a group of highly correlated variables have a tendency to be equal. In particular, if some variables are exactly equal, the regression method would assign identical coefficients.

Lemma 3.3.1. *Assume* $\mathbf{x}_i = \mathbf{x}_j$, *i, j* $\in \{1, \ldots, \text{p}\}$ *and let* $\lambda > 0$

1. *If J(·) is strictly convex, then* $\hat{\beta}_i = \hat{\beta}_j$
2. *If* $J(\beta) = \|\beta\|_1$, *then* $\hat{\beta}_i\hat{\beta}_j \geq 0$ *and* $\hat{\beta}^*$ *is another minimizer of* (3.6), *where*

$$\hat{\beta}^*_k = \begin{cases} \hat{\beta}_k \quad if \;\; k \neq i \;\; and \;\; k \neq j \\ (\hat{\beta}_i + \hat{\beta}_j)\cdot(s) \quad if \;\; k = i \\ (\hat{\beta}_i + \hat{\beta}_j)\cdot(1 - s) \quad if \;\; k = j \end{cases}$$

for any s $\in$ [0,1]

The proof of this Lemma is found in [2]. It shows a clear distinction between strictly convex and lasso penalties, which are convex but not strictly convex. The strictly convex penalty certifies the grouping effect when some variables are exactly equal. However the lasso in general does not have a unique solution. The naive elastic net penalty with $\lambda_2 > 0$ is strictly convex, so it has the assertion (1) property.

Theorem 3.3.1. *Given data* $(\mathbf{y}, \mathbf{X})$ *and parameters* (λ_1, λ_2)*, suppose the response* $\mathbf{y}$ *is centered and the predictors* $\mathbf{X}$ *are standardized. Let* $\hat{\beta}(\lambda_1, \lambda_2)$ *be the naive elastic net estimate. Suppose* $\hat{\beta}_i(\lambda_1, \lambda_2)\hat{\beta}_j(\lambda_1, \lambda_2) > 0$*. Let*

$$D_{\lambda_1,\lambda_2}(i,j) = \frac{1}{\|\mathbf{y}\|}\|\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)\|,$$

then $D_{\lambda_1,\lambda_2}(i,j) \leq \frac{1}{\lambda_2}\sqrt{2(1-\rho)}$*, where* $\rho = \mathbf{x}_i^T\mathbf{x}_j$*, the sample correlation.*

Proof

As $\hat{\beta}_i(\lambda_1, \lambda_2)\hat{\beta}_j(\lambda_1, \lambda_2) > 0$, then both $\hat{\beta}_i(\lambda_1, \lambda_2)$ and $\hat{\beta}_j(\lambda_1, \lambda_2)$ are non-zero. Moreover, sign($\hat{\beta}_i(\lambda_1, \lambda_2)$) = sign($\hat{\beta}_j(\lambda_1, \lambda_2)$). Since $\hat{\beta}(\lambda_1, \lambda_2)$ is the minimizer of (3.2),

$$\frac{\partial L(\lambda_1, \lambda_2, \beta)}{\partial \beta_k}|_{\beta=\hat{\beta}_{(\lambda_1,\lambda_2)}} = 0 \quad \text{if } \hat{\beta}_k(\lambda_1, \lambda_2) \neq 0 \tag{3.7}$$

Consequently,

$$-2\mathbf{x}_i^T(\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_1, \lambda_2)) + \lambda_1 sign(\hat{\beta}_i(\lambda_1, \lambda_2)) + 2\lambda_2\hat{\beta}_i(\lambda_1, \lambda_2) = 0 \tag{3.8}$$

$$-2\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_1, \lambda_2)) + \lambda_1 sign(\hat{\beta}_j(\lambda_1, \lambda_2)) + 2\lambda_2\hat{\beta}_j(\lambda_1, \lambda_2) = 0 \tag{3.9}$$

Substracting (3.9) from (3.8) gives:

$$(\mathbf{x}_j^T - \mathbf{x}_i^T)(\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_1, \lambda_2)) + \lambda_2(\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)) = 0$$

Because sign($\hat{\beta}_i(\lambda_1, \lambda_2)$) = sign($\hat{\beta}_j(\lambda_1, \lambda_2)$). It is equivalent to:

$$\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2) = \frac{1}{\lambda_2}(\mathbf{x}_i^T - \mathbf{x}_j^T)\hat{r}(\lambda_1, \lambda_2) \tag{3.10}$$

where $\hat{r}(\lambda_1, \lambda_2) = \mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_1, \lambda_2)$. As $\mathbf{X}$ is standardized, $\|\mathbf{x}_i^T - \mathbf{x}_j^T\|^2 = 2(1-\rho)$, $\rho = \mathbf{x}_i^T\mathbf{x}_j$. Since $\hat{\beta}$ is the minimizer of (3.2):

$$L(\lambda_1, \lambda_2, \hat{\beta}(\lambda_1, \lambda_2)) \leq L(\lambda_1, \lambda_2, \beta = 0),$$

$$\text{i.e, } \|\hat{r}(\lambda_1, \lambda_2)\|^2 + \lambda_2\|\hat{\beta}(\lambda_1, \lambda_2)\|^2 + \lambda_1\|\hat{\beta}(\lambda_1, \lambda_2)\|_1 \leq \|\mathbf{y}\|^2.$$

Then $\|\hat{r}(\lambda_1, \lambda_2)\| \leq \|\mathbf{y}\|$. Finally (3.10) implies:

$$D_{\lambda_1,\lambda_2}(i,j) \leq \frac{1}{\lambda_2}\frac{\|\hat{r}(\lambda_1,\lambda_2)\|}{\|\mathbf{y}\|}\|\mathbf{x}_j^T - \mathbf{x}_i^T\| \leq \frac{1}{\lambda_2}\sqrt{2(1-\rho)}$$

□

$D_{\lambda_1,\lambda_2}(i,j)$ shows the difference between the coefficients paths of predictors i and j. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are highly correlated, this theorem says that the difference between the coefficients paths of predictors i and j is about 0. This theorem also provides an upper bound which describes quantitatively the grouping effect of the naive elastic net.

The lasso does not have the grouping effect quality. The scenario where there is a group of variables among which the pairwise correlations are very high occurs frequently in practice, and lasso tends to select only one variable from the group and does not care which one select . In [7] the reader can find a theoretical explanation of this effect.

Chapter 4

Elastic Net

4.1 Deficiency of the naive elastic net

The naive elastic net overcomes some limitations of the lasso. However, empirical evidence shows that the naive elastic net does not perform correctly except if it is very close to either lasso or ridge. It is called *naive* for that reason.

The naive elastic net estimator is a two-phase process: Firstly, one finds the ridge regression coefficients for each fixed λ_2, subsequently one does the lasso shrinkage on the lasso coefficient solution paths. It appears to incur a double quantity of shrinkage. Double shrinkage does not help to reduce the variances much and adds non necessary extra bias, compared with ridge or lasso shrinkage. In this section we show an improvement of the prediction performance of the naive elastic net by correcting this double shrinkage.

4.2 The elastic net estimate

Given data ($\mathbf{y}$,$\mathbf{X}$), (λ_1,λ_2) as penalty parameters, and extended data ($\mathbf{y}^*$,$\mathbf{X}^*$), the naive elastic net solves a lasso type problem 3.2.1.

$$\hat{\beta}^* = \arg\min_{\beta^*} \left\{ \|\mathbf{y} - \mathbf{X}^*\beta^*\|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\|\beta^*\|_1 \right\} \tag{4.1}$$

The elastic net (rectified) estimates $\hat{\beta}$ are defined by:

$$\hat{\beta}(\text{elastic net}) = \sqrt{1+\lambda_2}\hat{\beta}^* \tag{4.2}$$

Recalling $\hat{\beta}$(naive elastic net) = $\frac{\lambda_1}{\sqrt{1+\lambda_2}}\hat{\beta}^*$, so :

$$\hat{\beta}(\text{elastic net}) = (1+\lambda_2)\hat{\beta}(\text{naive elastic net}) \tag{4.3}$$

Therefore, the elastic net is a rescaled naive elastic net coefficient. A scaling transformation is the simplest method to undo de shrinkage, moreover, it maintains the variable-selection property of the naive elastic net. Hence, the elastic net has all the good properties of the naive elastic net described in the previous chapter. Experimentally, it is seen that elastic net performs very well when is compared with ridge and lasso.

In this section we present a theoretical justification for choosing 1 + λ_2 as the scaling factor, considering the solution of the naive elastic net when the predictors are orthogonal.

A motivation for the (1 + λ_2) rescaling comes from a decomposition of the ridge operator. As $\mathbf{X}$ is standardized.

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & \rho_{12} & \cdot & \rho_{12} \\ & 1 & \cdot & \cdot \\ & & 1 & \rho_{p-1,p} \\ & & & 1 \end{bmatrix}_{p\times p},$$

where $\rho_{i,j}$ is the sample correlation between i and j predictors. Ridge estimates are given in (2.4), now it is considered $\lambda = \lambda_2$.

$\mathbf{R}$ can be rewritten as:

$$\mathbf{R} = \frac{1}{1+\lambda_2}\mathbf{R}^* = \frac{1}{1+\lambda_2}\begin{bmatrix} 1 & \frac{\rho_{12}}{1+\lambda_2} & \cdot & \frac{\rho_{12}}{1+\lambda_2} \\ & 1 & \cdot & \cdot \\ & & 1 & \frac{\rho_{p-1,p}}{1+\lambda_2} \\ & & & 1 \end{bmatrix}^{-1}\mathbf{X}^T \tag{4.4}$$

where $\mathbf{R}^*$ is like the usual OLS operator $\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right)$ except the correlations are shrunk by the factor $\frac{1}{1+\lambda_2}$, it will be called semi-correlation. Therefore, from (4.4) one can interpret that the ridge operator as semi-correlation followed by a scaling shrinkage.
When ridge's grouping effect is combined with the lasso, the $\frac{1}{1+\lambda_2}$ shrinkage step is not needed and removed by scaling. Although, ridge requires $\frac{1}{1+\lambda_2}$ shrinkage to control the estimation variance, in the new method, it is enough to rely on the lasso shrinkage to control the variance and obtain sparsity.

Let $\hat{\beta}$ = $\hat{\beta}$(elastic net). The next theorem gives another view of the elastic net,

Theorem 4.2.1. *Given data ($\mathbf{y}, \mathbf{X}$) and (λ_1, λ_2), then the elastic net estimates $\hat{\beta}$ are given by:*

$$\hat{\beta} = \arg\min_{\beta}\left\{\beta^T\left(\frac{\mathbf{X}^T\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2}\right)\beta - 2\mathbf{y}^T\mathbf{X}\beta + \lambda_1\|\beta\|_1\right\} \tag{4.5}$$

Proof

Let $\hat{\beta}$ the elastic net estimator. Using the definitions in (3.2.1) and (4.1),

$$\hat{\beta} = \arg\min_{\beta} \left\{ \left\| \mathbf{y}^* - \mathbf{X}^* \frac{\beta}{\sqrt{1+\lambda_2}} \right\|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}} \left\| \frac{\beta}{\sqrt{1+\lambda_2}} \right\|_1 \right\} =$$

$$\arg\min_{\beta} \left\{ \beta^t \left(\frac{\mathbf{X}^{*t}\mathbf{X}^*}{1+\lambda_2} \right) \beta - 2\frac{\mathbf{y}^{*t}\mathbf{X}^*}{\sqrt{1+\lambda_2}}\beta + \mathbf{y}^{*t}\mathbf{y}^* + \frac{\lambda_1\|\beta\|_1}{1+\lambda_2} \right\} \tag{4.6}$$

Replacing the indentities:

$$\mathbf{X}^{*t}\mathbf{X}^* = \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right)$$

$$\mathbf{y}^{*t}\mathbf{X}^* = \left(\frac{\mathbf{y}^t\mathbf{X}}{1+\lambda_2} \right)$$

$$\mathbf{y}^{*t}\mathbf{y}^* = \mathbf{y}^t\mathbf{y}$$

On the equation (4.6),

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{1+\lambda_2} \{ \beta^t \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right) \beta - 2\mathbf{y}^t\mathbf{X}\beta + \lambda_1\|\beta\|_1 \} + \mathbf{y}^t\mathbf{y} \right\} =$$

$$= \arg\min_{\beta} \left\{ \beta^t \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right) \beta - 2\mathbf{y}^t\mathbf{X}\beta + \lambda_1\|\beta\|_1 \right\}$$

□

When λ_2 = 0, Elastic net is equal to Lasso, so:

$$\hat{\beta}(\text{lasso}) = \arg\min_{\beta} \left\{ \beta^T(\mathbf{X}^T\mathbf{X})\beta - 2\mathbf{y}^T\mathbf{X}\beta + \lambda_1\|\beta\|_1 \right\} \tag{4.7}$$

Therefore, this theorem interprets the elastic net as a stabilized version of the lasso. Appreciate that $\hat{\Sigma} = \mathbf{X}^T\mathbf{X}$ is a sample version of the correlation matrix (Σ) and $\frac{\mathbf{X}^T\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2}$ = (1-γ)$\hat{\Sigma}$ + γ $\mathbf{I}$ with $\gamma = \frac{\lambda_2}{1+\lambda_2}$ shrinks $\hat{\Sigma}$ towards identity matrix. (4.5) and (4.7) say that rescaling after the elastic net penalization is equivalent to change $\hat{\Sigma}$ with its shrunk version in the lasso.
The lasso is a case of elastic net when λ_2 = 0. The other interesting case appears when $\lambda_2 \to \infty$:

$$\hat{\beta}(\infty) = \arg\min_{\beta} \beta^T\beta - 2\mathbf{y}^T\mathbf{X}\beta + \lambda_1\|\beta\|_1$$

$\hat{\beta}(\infty)$ has a closed form applying the KKT conditions:

$$\hat{\beta}(\infty)_i = \left(|\mathbf{y}^T\mathbf{x}_i| - \frac{\lambda_1}{2} \right)_+ \text{sgn}(\mathbf{y}^T\mathbf{x}_i), \quad i = 1, 2, \ldots, p \tag{4.8}$$

4.3 Elastic net in logit model

The Elastic net penalty can be applied to the logit model, similar to 2.2.1, when the lasso logit model is developed. Recalling the log-probabilistic function of the logit model (1.8)

$$\log L(\beta_0, \beta) = \sum_{i=1}^{n} [\mathbf{y}_i \log\{p(\mathbf{X}_i)\} + (1 - \mathbf{y}_i) \log\{1 - p(\mathbf{X}_i)\}]$$

The penalized form of the logit model using Elastic net penalization is:

$$\arg\max_{\beta} \left\{ \sum_{i=1}^{n} l(\beta) - (\alpha\|\beta\|^2 + (1-\alpha)\|\beta\|_1) \right\} \tag{4.9}$$

where $l(\beta)$ = log $L(\beta)$. The solution of (4.9) can be found by means of a Newton algorithm which can be found on [5]

4.4 Comparison between Lasso and Elastic net

It has been presented a new shrinkage method, but why is this method "better" than Lasso? Everybody knows that Lasso has good properties, but, for example, it does not perform satisfactorily when the predictors are highly correlated or the number of predictors (p) is much greater than the number of observations (n). In this section we present a different theoretical situation where Elastic net overcomes Lasso.

4.4.1 Introduction and Notation

Let p the number of predictors, and q the number of predictors with non-zero coefficients in the true linear model, p and q being fixed. Assume the data follow a regression model.

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{4.10}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$ is a vector of independent and identically distributed additive Gaussian noise with $\mu = 0$ and variance σ^2. Assume the first q elements of β are non-zeroes. Let $\beta_{(1)} = (\beta_1, \cdots, \beta_q)$ and $\beta_{(2)} = (\beta_{q+1}, \cdots, \beta_p)$, then $\beta_{(1)} \neq 0$ element-wise and $\beta_{(2)} = 0$.
Let $\mathbf{X}_1$ be the first q columns of $\mathbf{X}$, $\mathbf{X}_2$ the last $p - q$ columns and C(n) = $\frac{1}{n}\mathbf{X}^T\mathbf{X}$, in this section will be denoted C for simplicity. C can be expressed in block wise form:

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

where $C_{ij} = \frac{1}{n}\mathbf{X}_{(i)}^T\mathbf{X}_{(j)}$. It is shown when p and q are fixed, there is a condition called Irrepresentable Condition (IC) on the covariance matrix, which is necessary and sufficient for the Lasso's consistency. IC is presented on [8]

Irrepresentable condition. There exists a positive constant $\delta > 0$ (independent of n), with:

$$\|C_{21}C_{11}^{-1}(sign(\beta_{(1)})\|_\infty \leq 1 - \delta \tag{4.11}$$

In [9] the reader can find a necessary and sufficient condition for the Elastic Net when p and q are fixed. It is called Elastic Irrepresentable Condtition (EIC).

Elastic Irrepresentable condition. There exists a positive constant $\delta > 0$ (independent of n), with:

$$\left\| C_{21}\left(C_{11} + \frac{\lambda_2}{n}I\right)^{-1}\left(sign(\beta_{(1)}) + \frac{2\lambda_2}{\lambda_1}\beta_{(1)}\right)\right\|_\infty \leq 1 - \delta \tag{4.12}$$

Observe that EIC is IC when $\lambda_2 = 0$ and C_{11} is invertible. When C_{11} is invertible, λ_2 is fixed, λ_1 and n goes to ∞, the EIC reverts to the IC. If the Irrepresentable Condition holds, then there exist $\lambda_1 > 0$ and $\lambda_2 > 0$ which make Elastic Irrepresentable Condition hold. The results for the general scaling conditions of p, q and n are in [3].

4.4.2 Comparison

In this section Elastic net model selection is compared with that of the Lasso. Obviously, when Lasso selects the true model, Elastic net can also select the true model.

Proposition 4.4.1. *IC implies that for any $\lambda_1 > 0$, there exists λ_2, such that EIC holds, but the EIC does not imply IC.*

This result is trivial as $\lambda_2 = 0$ or small $\lambda_2 > 0$ leads EIC revert to IC: This proposition says that when IC holds, the EIC holds, so Elastic net can select the true model. A good question is what prior information is needed to suggest that the Elastic net selects the true model while Lasso does not? It is a hard question, but there are some situations when EIC holds and IC does not.

Consider the case when q (the number of predictors with non-zero coefficients) is equal to $p - 1$, so, there exists only one irrelevant predictor. Now a necessary and sufficient condition such that EIC holds is given.

First some regularity conditions are given on the model, they are easily satisfied.

$$0 < L_{min} \leq \Lambda(C_{11}) \leq L_{max}, \tag{4.13}$$

$$\|\beta\|_2 \geq c_1, \text{for a positive constant } c_1 > 0, \tag{4.14}$$

$$\|[C_{21}]_i\|_2 \geq c_2 \text{ for a positive constant } c_2 > 0 \text{ for } i = 1, \ldots, p-q, \tag{4.15}$$

where L_{min} and L_{max} are positive constants, $\Lambda(\cdot)$ means the eigenvalues of a matrix, and $[\cdot]_i$ denotes the ith row of a matrix. Consider β and C fixed.

Theorem 4.4.1. *Let* (4.13)*,* (4.14) *and* (4.15)*, and suppose that p - q = 1. When IC does not hold, for the sequence of λ_1 with $\lambda_1 \frac{\sqrt{q}}{n} \to 0$, there exists λ_2 that EIC holds when n is very big if and only if one of the two following conditions is satisfied:*

$$C_{21}C_{11}^{-1}sign(\beta_{(1)}) \geq 1 \ \ and \ \ C_{21}C_{11}^{-1}\beta_{(1)} < 0 \tag{4.16}$$

$$C_{21}C_{11}^{-1}sign(\beta_{(1)}) \leq -1 \ \ and \ \ C_{21}C_{11}^{-1}\beta_{(1)} > 0 \tag{4.17}$$

A proof of this theorem can be found in [3]

When $p - q \geq 2$ it is difficult to give a necessary and sufficient condition that EIC holds but (4.16) and (4.17) are necessary conditions for EIC hold. It is stated in this corollary:

Corollary 4.4.1. *Let* (4.13)*,* (4.14) *and* (4.15)*, and suppose that p - q > 1. When IC does not hold, for the sequence of λ_1 with $\lambda_1 \frac{\sqrt{q}}{n} \to 0$, there exists λ_2 that EIC holds when n is very big only if, for all i = 1,. . ., p - q*

$$[C_{21}]_i C_{11}^{-1}\beta_{(1)} < 0 \ \text{ when } \ [C_{21}]_i C_{11}^{-1}sign(\beta_{(1)}) \geq 1 \tag{4.18}$$

$$[C_{21}]_i C_{11}^{-1}\beta_{(1)} > 0 \ \text{ when } \ [C_{21}]_i C_{11}^{-1}sign(\beta_{(1)}) \leq -1 \tag{4.19}$$

Chapter 5

Elastic net in R studio

In this chapter we will describe the version 1.3 of the package $elasticnet$, which was created by Zou and Hastie in 2020, and the most important commands of the package $glmnet$, which was created by Friedman, Zou, Hastie ... in 2021. The $elasticnet$ package uses some functions from the package $lars$, therefore, before using it, one should install $lars$. Those packages provide functions for fitting the solution path of the Elastic net.

Before showing the package, a brief explanation about the Least Angle regression (LAR) algorithm is shown as it is needed in $elasticnet$. Least angle regression uses a similar strategy to that employed by forward stepwise regression, but it incorporates the predictors gradually, so each predictor takes part in the model as much as it "deserves". In the first step, it identifies the predictor most correlated with the response. Instead of adjusting this predictor completely, LAR moves the coefficient of this predictor continuously towards its least squares value. When another variable "catches up" in terms of correlation with the residual, the process stops. The second variable then joins the active set, and their coefficients move together so that their correlations move simultaneously, decreasing in value. This process continues until all variables are in the model, and ends with the full least squares fit.

5.1 Elasticnet package

cv.enet
This function calculates the K-fold cross validated mean squared prediction error for elastic net.

Usage

$$cv.enet(x, y, K = 10, lambda, s, mode, trace = FALSE, plot.it = TRUE, se = TRUE, \ldots)$$

Arguments

x	Input to lars
y	Input to lars
K	Number of folds (10 predefined)
lambda	Elastic net penalty parameter
s	Abscissa value at which the cross validation curve should be computed. Its values depends on the mode = argument
mode	mode $\in$ ("step", "fraction", "norm", "penalty"). If mode = "step", the s = argument indexes the LARS step number. When mode ="fraction", then s is a number between 0 and 1, it refers to the ratio of the L1 norm of the coefficients vector. If mode = "norm", s refers to the L1 norm of the coefficient vector. If mode = "penalty", s should be the 1-norm penalty parameter
trace	Show computations? (FALSE default)
plot.it	Plot it? (TRUE default)
$\cdots$	Additional arguments to enet

Values

Invisibly returns a list with components (one can plot this with plotCVLars), this list is:

fraction	The values of s
cv	The CV curve at each value of s
cv.error	The standard error of CV curve

enet
Fits Elastic Net regression models using the LARS algorithm which computes the complete elastic net solution for all values of the shrinkage parameter. It has the same computational cost as the least square fit.

Usage

$$enet(x, y, lambda, max.steps, normalize = TRUE, intercept = TRUE, trace = FALSE, eps = .Machine\$double.eps)$$

Arguments

x	Matrix of the predictors.
y	response.
lambda	Elastic net penalty parameter. For lambda = 0 do the Lasso fit.
max.steps	The maximum steps the function can take. 50* min(p,$n-1$), where p is the number of variables, and n is the number of samples. It can be used to perform early sttoping
normalize	Standardize the predictors?(TRUE default)
intercept	Center the predictors? (TRUE default)
trace	Show computations? (FALSE default)
eps	An effective zero

Values
$enet$ returns an object which can be printed, plotted and predicted

plot.enet
Produces a plot of an enet fit.

Usage

$$plot(x, xvar = c("fraction","penalty","L1norm","step"), use.color = FALSE, \ldots)$$

Arguments

x	An enet object.
xvar	The class of x against which to plot. If xvar= " fraction" (default) it plots against the fraction of the L1 norm. of the coefficient vector. If xvar = "penalty" it plots against the 1-norm penalty parameter. If xvar = "L1norm" (also can be written as xvar = "L1") it plots against the L1 norm of the coefficient vector. Finally, if xvar = "step" it plots against the LARS step number.
use.color	Use color on the plot? (FALSE default)
...	Arguments for a generic plot

Value

NULL

---

predict.enet
Makes predictions from a fitted elastic net model, with this function one is allowed to extract a prediction at a particular point on the path using the LARS algorithm, as long as, $enet()$ produces the entire path.

Usage

$$predict(object, newx, s, type = c("fit","coefficients"), mode = c("step","fraction","norm","penalty"), naive = FALSE, ...)$$

Arguments

object	A fitted enet object.
newx	It depends on the value of type. When type = "fit", newx should be the values of x at which the fit is required. Otherwise, when type="coefficient", newx can be ommited.
s	A value, or a vector of values which index the path. It depends on mode argument (mode = "step" default).
type	If type = "fit", it returns the fitted values. If type = "coefficients", it returns the coefficients
mode	mode $\in$ ("step", "fraction", "norm", "penalty") If mode = "step", the s = argument indexes the LARS step number, the coefficients will be returned corresponding to the values corresponding to step s . When mode ="fraction", then s is a number between 0 and 1, it refers to the ratio of the L1 norm of the coefficients vector, relative to the norm at the full solution. If mode = "norm", s refers to the L1 norm of the coefficient vector. If mode = "penalty", s should be the 1-norm penalty parameter. .
naive	Naive elastic net fit? (FALSE default).
...	Arguments for a generic plot.

Value

A vector/matrix of coefficients, or a vector/matrix of fitted values

---

print.enet
Prints method for enet objects.

Usage

$$print(x, ...)$$

Arguments

x	An enet object.
...	Arguments for a generic plot.

Value

NULL

5.2 glmnet package

The $glmnet$ package is more general than $elasticnet$, e.g. logistic regression can be performed. However, it is more difficult to use, so we will overview only the most important commands of this package.

---

glmenet
fits a generalized linear model with elastic net regularization

---

Usage

$$glmnet(x, y, family = c(\text{"}gaussian\text{"}, \text{"}binomial\text{"}), alpha = 1, \ldots)$$

Arguments

x	input matrix
y	response vector, quantitative if family = "gaussian", a factor with two levels if family = "binomial" (logit)
alpha	elastic net penalty. Alpha = 1 gives lasso penalty and alpha = 0 ridges penalty.
...	other arguments

Values

An object with S3 class "glmenet", "enet" or "lognet" depending on family value. The object can be printed, predicted or plotted with the commands print predict and plot.

beta	The coefficients matrix
...	other values

Chapter 6

Numerical examples

In this chapter, some examples will be displayed, a simulation case and two database experiments.

6.1 Prostate Database

6.1.1 Description

Name: Prostate, [1] .
In this dataframe there are 97 observations of prostate-specific antigen(psa) and some clinical measures in men, in order to fit a linear model to the psa.

Predictors

- lcavol: logarithm of cancer volume
- lweight: logarithm of prostate weight
- age: age of the man
- lbph: logarithm of the amount of capsular hyperplasia
- svi: seminal vesicle invasion
- lcp: logarithm of capsular penetration
- gleason: Gleason score
- pgg45: percent of Gleason scores 4 or 5

Response

- lpsa: logarithm of psa

6.1.2 Ordinary Least Square regression

In this section, an OLS regression will be done to the database described in the previous section. After loading the data with the commands:

prostate= read.csv$("prostate.txt", sep = "", dec = ".", header = TRUE)$
prostate=prostate[,2:10]

Now the OLS regression is done with this command:
ols <- **lm** (lpsa ., data = prostate)
After running **summary**(ols), the coefficients and some information are shown on R console as follows.

```
Call:
lm(formula = lpsa ~ ., data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.76644 -0.35510 -0.00328  0.38087  1.55770

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.181561   1.320568   0.137  0.89096
lcavol       0.564341   0.087833   6.425 6.55e-09 ***
lweight      0.622020   0.200897   3.096  0.00263 **
age         -0.021248   0.011084  -1.917  0.05848 .
lbph         0.096713   0.057913   1.670  0.09848 .
svi          0.761673   0.241176   3.158  0.00218 **
lcp         -0.106051   0.089868  -1.180  0.24115
gleason      0.049228   0.155341   0.317  0.75207
pgg45        0.004458   0.004365   1.021  0.31000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6995 on 88 degrees of freedom
Multiple R-squared:  0.6634,    Adjusted R-squared:  0.6328
F-statistic: 21.68 on 8 and 88 DF,  p-value: < 2.2e-16
```

Figure 6.1: Summary of the OLS model

Observations:
In $Figure\ 6.1$, the $Pr(>|t|)$ column shows the p-values for the next hypothesis test:

$$\begin{cases} H_0 : \beta_i = 0 \\ H_1 : \beta_i \neq 0 \end{cases} \tag{6.1}$$

Therefore, one can conclude that the variables for which the null hypothesis is rejected are **lcavol**, **lweight**, **lbph**, **svi** and **age**. Conversely, the variables for which the null

hypothesis is plausible are **lcp**, **gleason** and **pgg45**, so those variables might not be significant on the OLS regression model.

Moreover, one can appreciate that the R^2 coefficient is equal to 0.65, so the model may approximate slightly good the dataset.

Now a set of figures are going to be displayed, they summarize the OLS model running the command **plot**(ols).

The $Figure$ 6.2 shows fitted values versus residuals; one can observe that the red line is almost $\mathbf{y} = 0$, that means that the linear fit is good. Moreover, one can see three distiguished observations: **47**, **95** and **39**.

nn $Figure$ 6.3 it has been done a normality analysis via a Q-Q plot in order to see if the data might follow a normal distribution, as the data almost fits to the discontinuous line, except at the start and at the end because of the outliers, one could conclude that the data follows a normal distribution.

The $Figure$ 6.4 shows a homocedasticity test, as the red line in the figure fits to an horizontal line, one could accept the homocedasticity.

The $Figure$ 6.5 is useful to detect the outliers (aberrant observations), **69**, **95** and **47** they will not be equal as the three distiguished observations of the $Figure$ 6.2. Observe that **69** and **47** are distiguished and outliers, so if one repeats the OLS regression removing the outliers it will fit better.

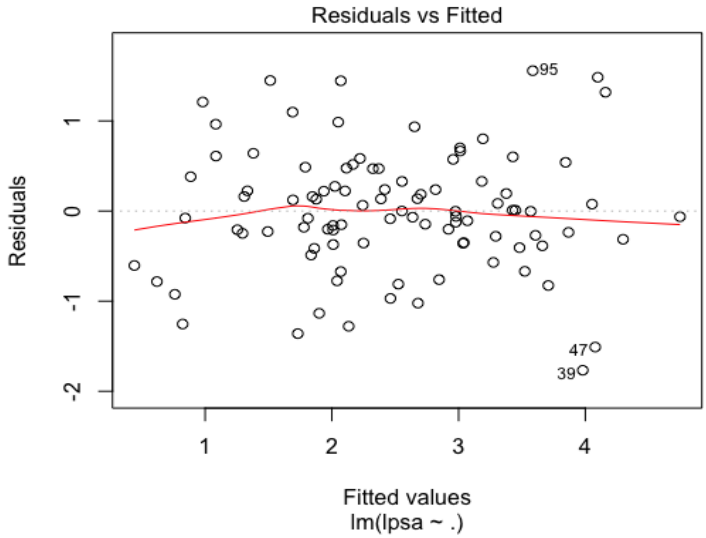

Figure 6.2: How good is the linear model?

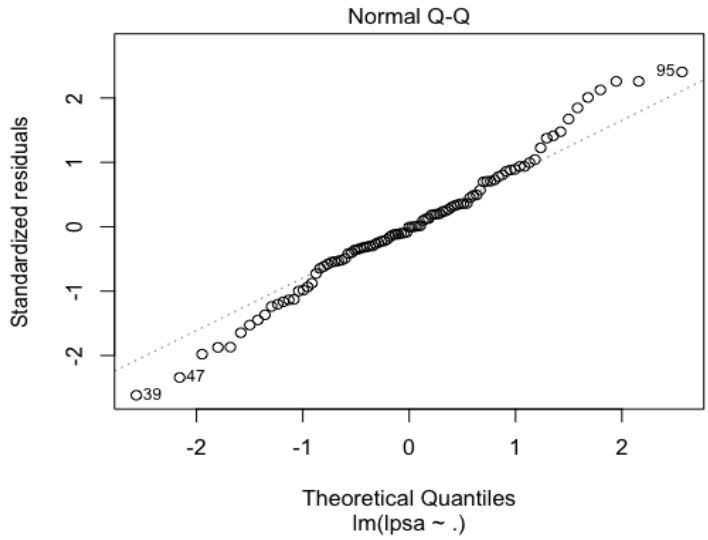

Figure 6.3: Normality test

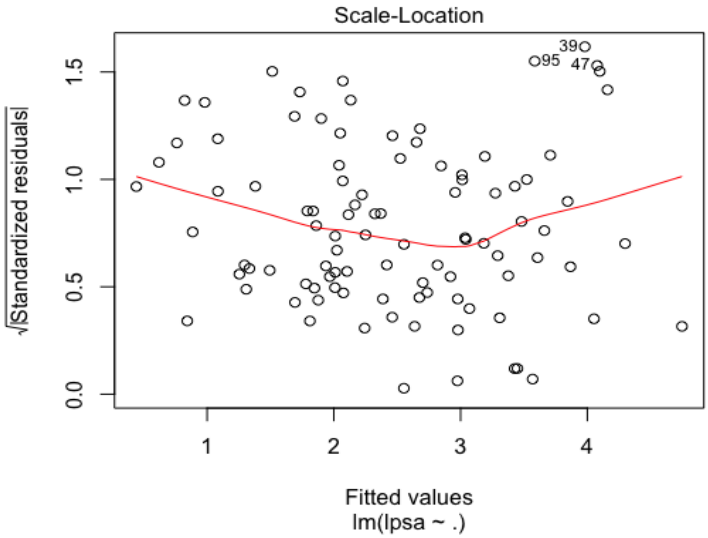

Figure 6.4: Homoscedasticity test

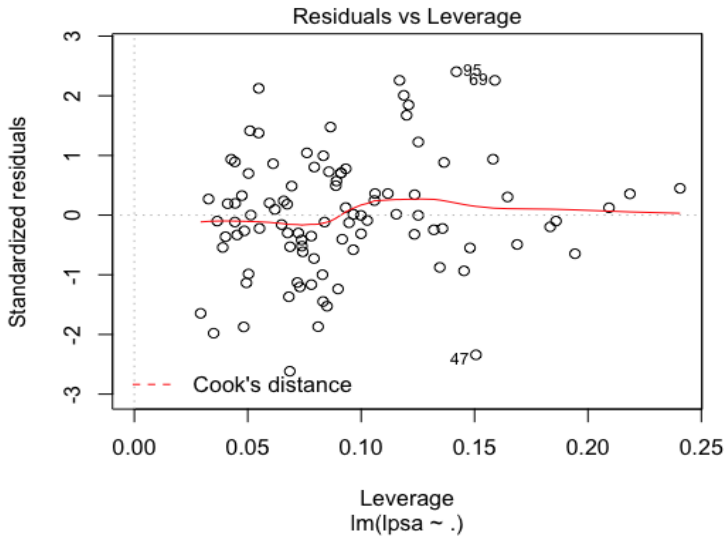

Figure 6.5: Outliers?

6.1.3 Lasso regression model

In this section the Lasso method is going to be used. At first one has to load the $lars$ package with the command **library(lars)**. The data must be in the class **X** as matrix and **y** as numeric in order to use the $lars$ functions, so using this command the data has the correct class. **X = as.matrix(prostate[,1:8])** and **y =prostate[,9]**.

Using the following command the Lasso regression is done.

lasso < - lars(X,y,type=c("lasso"))

With the following command the Lasso coefficient path is obtained.

plot(lasso)

In the $Figure$ 6.7 there is an image in which the lasso regression method is done. From the right to the left, one can see that the predictor's coefficients are shrinking to zero in every algorithm step.

In the $Figure$ 6.6 we display the coefficients of the method in every step of the algorithm. Observe that in every step one coefficient becomes a non-zero coefficient. The first variable for which its coefficient becomes zero is **lcp**, so for this algorithm **lcp** is the least significant variable. However, on $Figure$ 6.1 one can see that the least

significant variable is **gleason** on the OLS model.

In the $Figure\ 6.8$ we show the first 15 predictions obtained with the Lasso regression. In the first column, the Lasso model does not have any variable, on the second column, the model has one variable, then successively.

On the R-studio object lasso, defined before, one can extract the value of $\lambda = (\lambda_1, \ldots, \lambda_8)$, where λ_i is the value of λ on the step i. $\lambda_{intercept} \to \infty$ is the value for which all predictors are equal to 0 on the model. The following table is a three-column table. The first column displays the different values of λ. The second column displays, the proportions between λ_1 and the others values of λ. Finally on the third column, the mean squared error (MSE) of the predictions for every value of λ is given.

λ_i values	$\frac{\lambda_i}{\lambda_1}$ proportions	MSE
$\lambda_{intercept} = \infty$	-	1.318739
λ_1 = 8.3067969	1	0.7875459
λ_2 = 4.1805708	0.50327110	0.7241959
λ_3 = 3.5705887	0.42983942	0.5179826
λ_4 = 1.4068330	0.16935926	0.5078044
λ_5 = 1.2293599	0.14799446	0.4774012
λ_6 = 0.6286376	0.07567750	0.4600064
λ_7 = 0.3630874	0.04370967	0.4541516
λ_8 = 0.2164061	0.02605169	0.4439012

Table 6.1: λ, proportions and MSE

Observe that the model with least MSE is the model with all variables. However, there are slightly difference between the MSE of the model with 5 variables and that of the model with all variables.

```
       lcavol   lweight         age       lbph        svi        lcp    gleason        pgg45
[1,] 0.0000000 0.0000000  0.000000000 0.00000000 0.00000000  0.0000000 0.00000000 0.0000000000
[2,] 0.3573072 0.0000000  0.000000000 0.00000000 0.00000000  0.0000000 0.00000000 0.0000000000
[3,] 0.3916323 0.0000000  0.000000000 0.00000000 0.09772183  0.0000000 0.00000000 0.0000000000
[4,] 0.4729686 0.4010287  0.000000000 0.00000000 0.44189300  0.0000000 0.00000000 0.0000000000
[5,] 0.4772307 0.4343405  0.000000000 0.00000000 0.46205106  0.0000000 0.00000000 0.0003752489
[6,] 0.4945025 0.4904080  0.000000000 0.03525646 0.55370843  0.0000000 0.00000000 0.0013866469
[7,] 0.5067509 0.5431376 -0.008040524 0.06070074 0.58841287  0.0000000 0.00000000 0.0022898666
[8,] 0.5115481 0.5748987 -0.012590999 0.07452697 0.61037529  0.0000000 0.01704893 0.0024820450
[9,] 0.5643413 0.6220198 -0.021248185 0.09671252 0.76167340 -0.1060509 0.04922793 0.0044575118
```

Figure 6.6: Lasso coeffcients

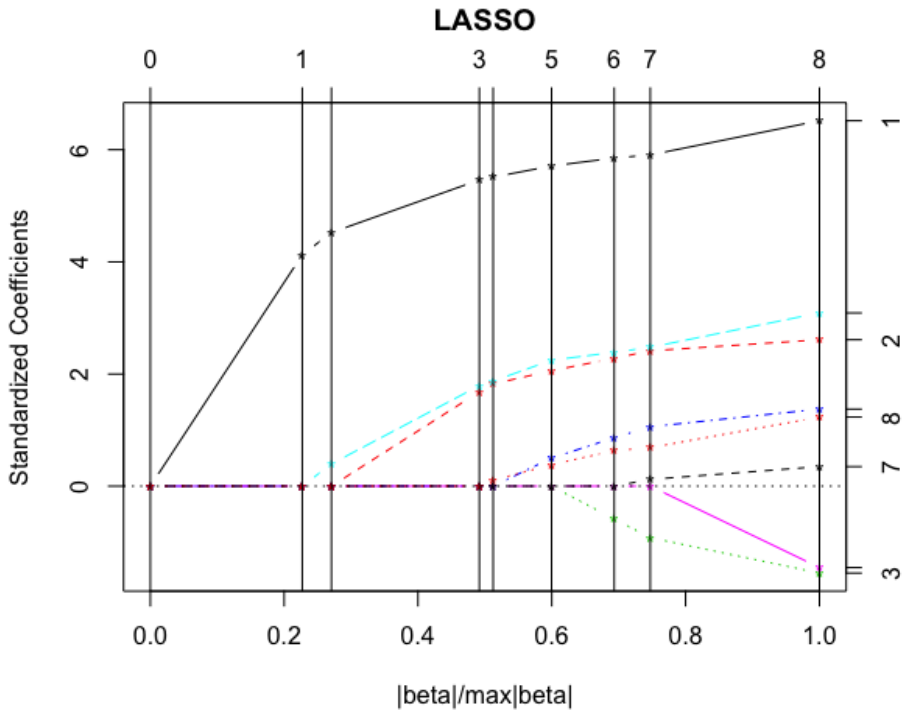

Figure 6.7: Lasso regression

```
$fit
       [,1]     [,2]     [,3]     [,4]     [,5]      [,6]      [,7]      [,8]      [,9]
1  2.478387 1.788845 1.701448 1.125294 1.074924 0.8964870 0.8716574 0.8553725 0.8229078
2  2.478387 1.640765 1.539142 1.149912 1.116104 0.9613547 0.8961349 0.8589320 0.7612550
3  2.478387 1.813497 1.728468 1.126558 1.081382 0.9199793 0.7169623 0.6102052 0.4416131
4  2.478387 1.565831 1.457009 1.035948 1.000019 0.8395823 0.7698512 0.7304723 0.6199877
5  2.478387 2.264505 2.222802 2.020773 1.998161 1.8798841 1.8098288 1.7663793 1.7315458
6  2.478387 1.620910 1.517379 1.087216 1.050146 0.8893463 0.8829821 0.8790325 0.8434007
7  2.478387 2.259413 2.217221 2.030533 2.009230 1.9635794 1.9303643 1.9067252 1.9001676
8  2.478387 2.243685 2.199982 2.036179 2.016887 2.0066707 2.0480908 2.0663827 2.1330020
9  2.478387 1.718559 1.624410 1.341068 1.315512 1.1768519 1.2143389 1.2352194 1.2546269
10 2.478387 2.075750 2.015914 1.695592 1.664471 1.5265392 1.4320688 1.3755689 1.2953383
11 2.478387 2.087004 2.028249 1.854698 1.835689 1.7184631 1.6272587 1.5732301 1.4942925
12 2.478387 1.514700 1.400966 1.094948 1.068931 1.0172784 0.9897587 0.9736583 0.8860818
13 2.478387 2.572509 2.560395 2.264253 2.242930 2.1469235 2.0848892 2.0508322 2.0413797
14 2.478387 2.523779 2.506983 2.189871 2.157765 2.0327368 1.9129906 1.8544907 1.8362787
15 2.478387 2.426921 2.400820 2.239632 2.221154 2.1163262 2.0970659 2.0968656 2.0704933
```

Figure 6.8: Lasso predictions

6.1.4 Elastic Net Regression

In this section we show how to run Elastic net regression in R. The package $elasticnet$ is needed, so one starts loading the library with the command **library(elasticnet)**, as in lasso regression the data must be on the same class, **X = as.matrix(prostate[,1:8])** and **y =prostate[,9]**.

Using the next command the Elastic net regression is done.

elastic < - enet(X,y,lambda = λ**)**

Depending on the value of λ one has a different elastic net regression, for λ = 0 the Lasso is obatined. One can see that $Figure$ 6.7 and $Figure$ 6.9 are the same plot. This is obvious as elastic net is lasso when λ = 0. Other interesting case is when λ = 1, where the ridge regression is obtained. In $Figure$ 6.10 the reader can find the coefficient paths of the elastic net when λ = 1.

The following table gives the MSE table, where the rows are the different values of λ and the columns are the number of predictors are used in the model.

	0	1	2	3	4	5	6	7	8
0	1.319	0.788	0.724	0.518	0.508	0.477	0.460	0.454	0.443
0.1	1.319	0.823	0.728	0.531	0.498	0.473	0.470	0.453	0.451
0.2	1.319	0.850	0.730	0.551	0.541	0.493	0.481	0.471	0.464
0.3	1.319	0.871	0.732	0.614	0.542	0.492	0.488	0.479	0.480
0.4	1.319	0.888	0.733	0.669	0.545	0.494	0.491	0.492	0.500
0.5	1.319	0.901	0.734	0.715	0.549	0.497	0.492	0.508	0.522
0.6	1.319	0.912	0.748	0.734	0.552	0.500	0.494	0.529	0.546
0.7	1.319	0.922	0.770	0.733	0.556	0.503	0.497	0.552	0.571
0.8	1.319	0.930	0.788	0.732	0.560	0.506	0.501	0.578	0.598
0.9	1.319	0.937	0.803	0.730	0.563	0.508	0.505	0.606	0.626
1	1.319	0.943	0.816	0.729	0.567	0.510	0.510	0.637	0.655

Table 6.2: MSE for different values of λ on elastic net regression

In this table, one can appreciate that the minimum value of the MSE is for λ = 0 and 8 predictors, this means that the best model is the LASSO because λ = 0 , but this would not be true if the data were divided into a training set (67 observations) and a test set (30 observations). It is obtained that the lasso performs much better than OLS, moreover elastic net performs better than lasso, see [2].

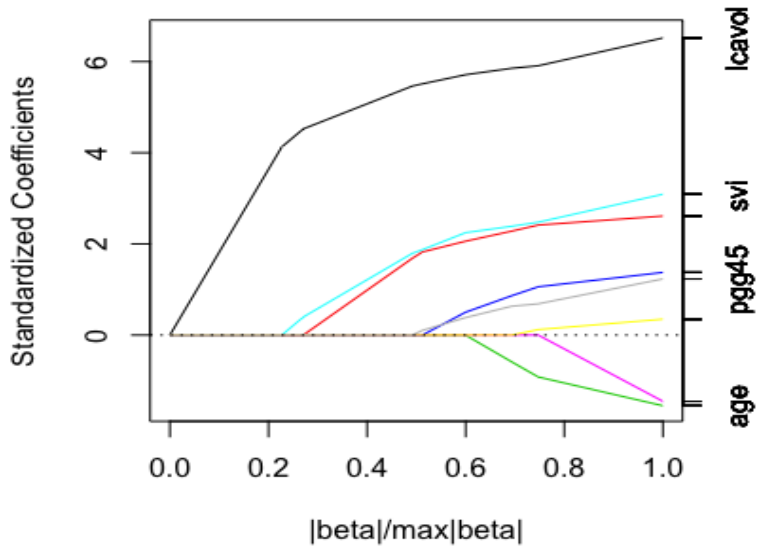

Figure 6.9: Elastic net regression. $\lambda = 0$

6.2 Hitters database

6.2.1 Description

Name: Hitters, this database is included in the package $ISLR$.
This data frame has 322 observations of some stats from the players of the Major League Baseball. There are 19 predictors and the response is the salary of the player (to see more information of the predictors see [10]). This database has some observations with a Na value, so before doing the regression one has to eliminate such observations. To load the database it is used the command:

data(Hitters, package = "ISLR")
Hitters = na.omit(Hitters)

6.2.2 Ordinary Least Square regression

In this section, an OLS regression will be done to the database described in the previous section. It is done with the following command:
ols <- **lm** (Salary ., data = Hitters)
After running **summary**(ols), the coefficients and some information are shown in the R console, see $Figure$ 6.11.

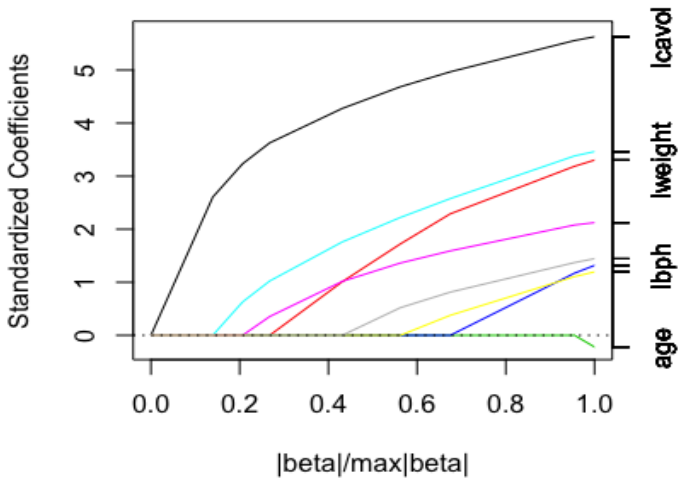

Figure 6.10: Elastic net regression. $\lambda = 1$

```
Call:
lm(formula = Salary ~ ., data = Hitters)

Residuals:
    Min      1Q  Median      3Q     Max
-907.62 -178.35  -31.11  139.09 1877.04

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  163.10359   90.77854   1.797 0.073622 .
AtBat         -1.97987    0.63398  -3.123 0.002008 **
Hits           7.50077    2.37753   3.155 0.001808 **
HmRun          4.33088    6.20145   0.698 0.485616
Runs          -2.37621    2.98076  -0.797 0.426122
RBI           -1.04496    2.60088  -0.402 0.688204
Walks          6.23129    1.82850   3.408 0.000766 ***
Years         -3.48905   12.41219  -0.281 0.778874
CAtBat        -0.17134    0.13524  -1.267 0.206380
CHits          0.13399    0.67455   0.199 0.842713
CHmRun        -0.17286    1.61724  -0.107 0.914967
CRuns          1.45430    0.75046   1.938 0.053795 .
CRBI           0.80771    0.69262   1.166 0.244691
CWalks        -0.81157    0.32808  -2.474 0.014057 *
LeagueN       62.59942   79.26140   0.790 0.430424
DivisionW   -116.84925   40.36695  -2.895 0.004141 **
PutOuts        0.28189    0.07744   3.640 0.000333 ***
Assists        0.37107    0.22120   1.678 0.094723 .
Errors        -3.36076    4.39163  -0.765 0.444857
NewLeagueN   -24.76233   79.00263  -0.313 0.754218
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 315.6 on 243 degrees of freedom
Multiple R-squared:  0.5461,    Adjusted R-squared:  0.5106
F-statistic: 15.39 on 19 and 243 DF,  p-value: < 2.2e-16
```

Figure 6.11: Summary of the OLS model

Observations:
In $Figure\ 6.11$, the $Pr(>|t|)$ column shows the p-values for the following hypothesis test:

$$\begin{cases} H_0 : \beta_i = 0 \\ H_1 : \beta_i \neq 0 \end{cases} \tag{6.2}$$

Therefore, one can conclude that the variables for which the null hypothesis is rejected are **AtBat**, **Hits**, **Walks**, **Cwalks**, **PutOuts** and **DivisionW**. Conversely, the null hypothesis is plausible for the other variables, so those variables might not be significant on the OLS regression model.

Moreover, one can appreciate that the R^2 coefficient is equal to 0.51, so the model may approximate mildly good the dataset.

Now a set of figures are going to be displayed, they summarize the OLS model running the command **plot**(ols).

The $Figure\ 6.12$ shows fitted values versus residuals; one can observe that the red line is almost $\mathbf{y} = 0$, that means that the linear fit is good. Moreover, one can see three distiguished observations: **Mike Schimidt**, **Ozzie Smith** and **Reggie Jackson**.

In $Figure\ 6.13$ it has been done a normality analysis via Q-Q plots in order to see if the data might follow a normal distribution, as the data almost fits to the discontinuous line, except at the start and at the end because of the outliers, one could conclude that the data follows a normal distribution.

The $Figure\ 6.14$ shows a homocedasticity test. Observe that the red line in the figure increases if the x increases, so it does not fit to an horizontal line and the homocedasticity is rejected.

The $Figure\ 6.15$ is useful to detect the outliers (aberrant observations), **Mike Schimidt**, **Reggie Jackson** and **Pete Rose** they will not be equal as the three distiguished observations of the $Figure\ 6.12$. Observe that **Mike Schimidt** and **Reggie Jackson** are distiguished and outliers, so if one repeat the OLS regression eliminating the outliers it will fit better.

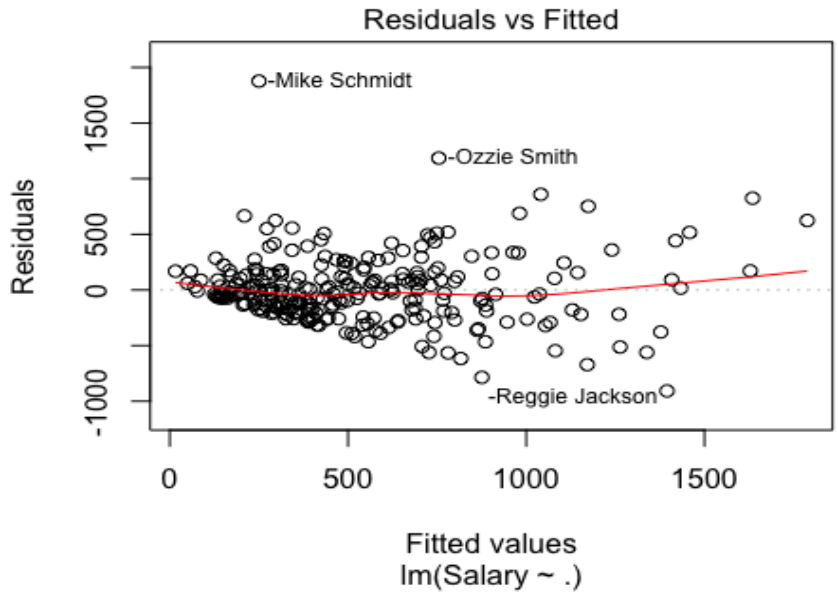

Figure 6.12: How good is the linear model?

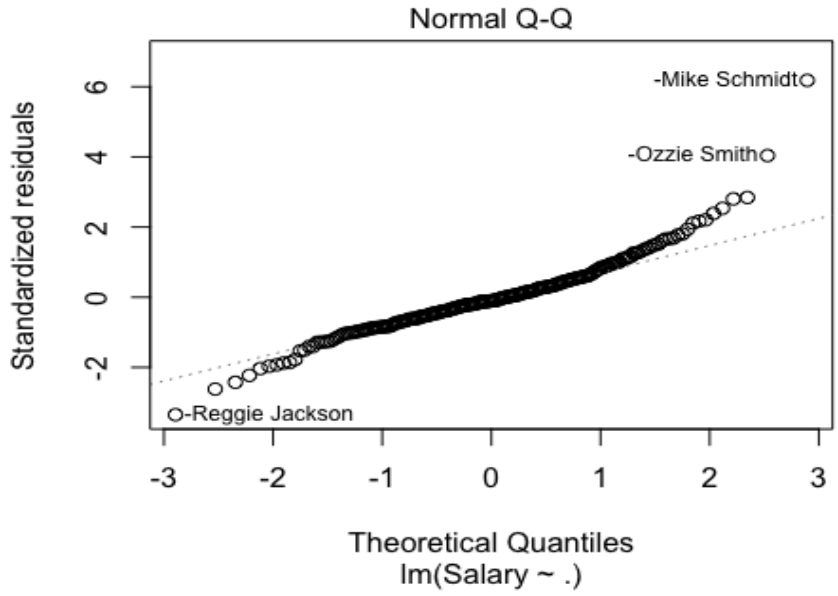

Figure 6.13: Normality test

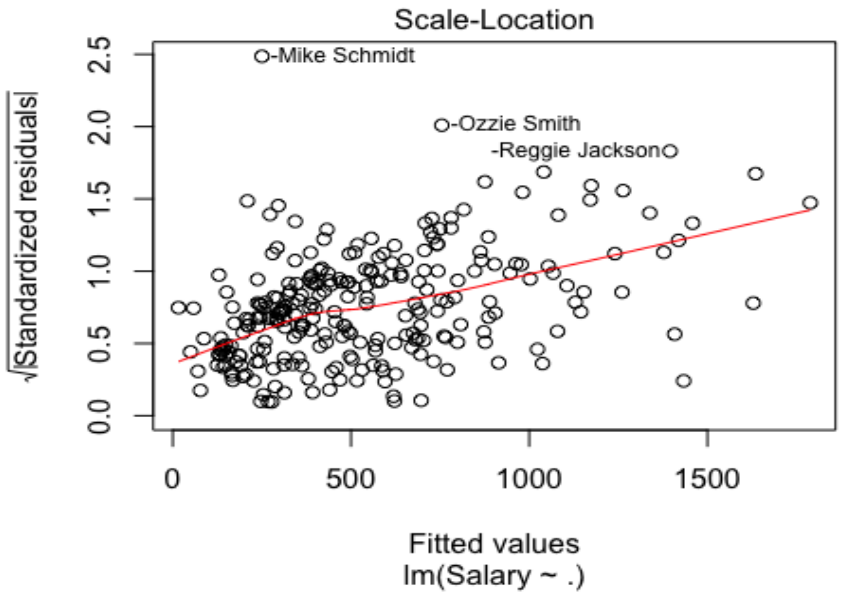

Figure 6.14: Homoscedasticity test

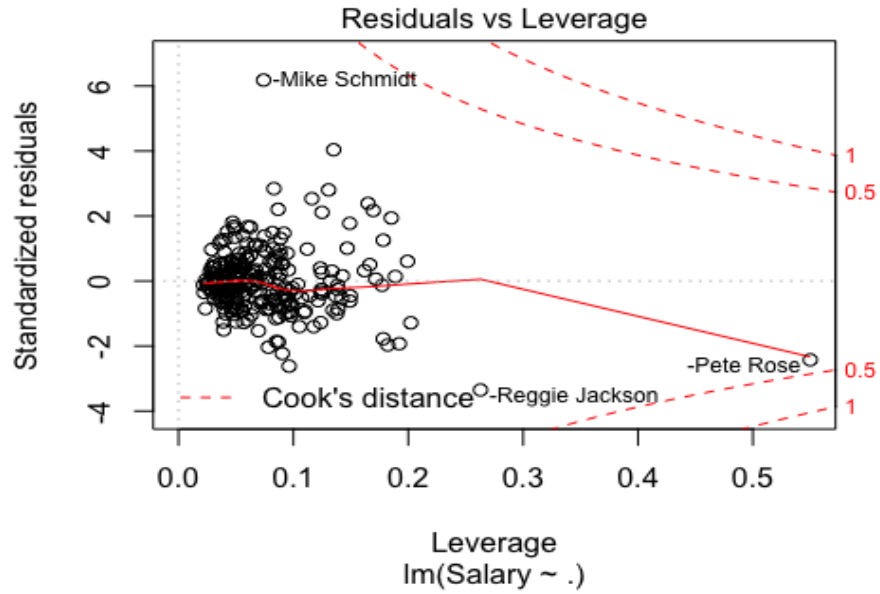

Figure 6.15: Outliers?

6.2.3 Lasso regression model

In this section the lasso is used. At first one has to load the $lars$ package with the command **library(lars)**. The data must be in the class $\mathbf{X}$ as matrix and $\mathbf{y}$ as numeric in order to use the $lars$ functions, so using this command the data has the correct class. **X = as.matrix(Hitters[,-19])** and **y =HItters[,19]**.

Using the following command the Lasso regression is done.

lasso < - lars(X,y,type=c("lasso"))

With the following command the Lasso coefficient path is obtained.

plot(lasso)

$Figure$ 6.16 there is an image in which one can see the coefficient paths. From the right to the left, one can see that the predictor's coefficients are shrinking into zero in every algorithm step.

$Figure$ 6.17 displays the coefficients of the method in the first 9 steps of the algorithm. Observe that in every step one coefficient becomes a non-zero coefficient. The first variable for which its coefficient becomes non-zero is **CRBI**, so for this algorithm **CRBI** is the most significant variable.

$Figure$ 6.18 shows the first 15 predictions obtained with the Lasso regression. In the first column, the Lasso model does not have any variable, on the second column, the model has one variable, then successively.

On the R-studio object lasso, defined before, one can extract the value of $\lambda = (\lambda_1, \ldots, \lambda_{20})$, where λ_i is the value of λ on the step i. $\lambda_{intercept} \to \infty$ is the value for which all predictors are equal to 0 in the model. The following table is a three-column table for which it is shown the 10 first rows. The first column displays the different values of λ. The second column displaying the ratio between λ_1 and the remaining λ values. Finally, the third column displays the mean squared error (MSE) of the predictions for every value of λ.

λ_i values	$\frac{\lambda_i}{\lambda_1}$ proportions	MSE
$\lambda_{intercept} = \infty$	-	202734.3
$\lambda_1 = 4139.979895$	1	185851.4
$\lambda_2 = 3563.598481$	0.8607767601	169696.3
$\lambda_3 = 2926.749171$	0.7069476773	159221.2
$\lambda_4 = 2625.666890$	0.6342221355	131235.2
$\lambda_5 = 1643.051047$	0.3968741610	120530.6
$\lambda_6 = 1197.966217$	0.2893652258	104888.7
$\lambda_7 = 284.87903$	0.0688116940	104370.5
$\lambda_8 = 221.627467$	0.0535334645	104189.9
$\lambda_9 = 203.185815$	0.0490789377	103683
...	...	...

Table 6.3: λ, proportions and MSE

From 6.3 one can deduce that the more variables the model has, the better the fit (in the training sample). However looking at MSE and values of λ a model with 6 variables will be a really good model and the difference between the MSE is so small, moreover the proportion becomes so small from λ_7. On *Figure* 6.17 there is the order for which the predictors are included in the model. So for λ_6 the predictors that are used are: **CRBI**, **CRuns**, **Hits**, **Walks**, **PutOuts** and **DivisionW**

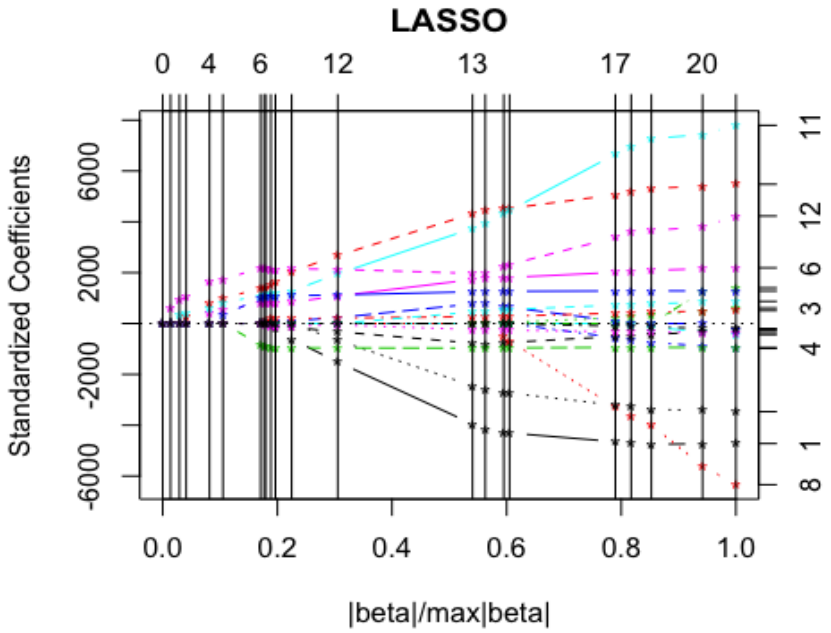

Figure 6.16: Lasso regression

	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	LeagueN	DivisionW	PutOuts	Assists	Errors	NewLeagueN
[1,]	0	0.000	0	0	0	0.000	0	0	0	0	0.000	0.000	0	0.000	0.000	0.000	0	0.000	0
[2,]	0	0.000	0	0	0	0.000	0	0	0	0	0.000	0.110	0	0.000	0.000	0.000	0	0.000	0
[3,]	0	0.000	0	0	0	0.000	0	0	0	0	0.061	0.173	0	0.000	0.000	0.000	0	0.000	0
[4,]	0	0.334	0	0	0	0.000	0	0	0	0	0.076	0.205	0	0.000	0.000	0.000	0	0.000	0
[5,]	0	1.124	0	0	0	1.164	0	0	0	0	0.112	0.310	0	0.000	0.000	0.000	0	0.000	0
[6,]	0	1.386	0	0	0	1.530	0	0	0	0	0.153	0.333	0	0.000	0.000	0.072	0	0.000	0
[7,]	0	1.860	0	0	0	2.213	0	0	0	0	0.204	0.414	0	0.000	-102.149	0.219	0	0.000	0
[8,]	0	1.912	0	0	0	2.254	0	0	0	0	0.207	0.421	0	8.760	-109.098	0.227	0	0.000	0
[9,]	0	1.941	0	0	0	2.260	0	0	0	0	0.208	0.421	0	11.768	-111.045	0.230	0	-0.249	0

Figure 6.17: Lasso coeffcients

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
-Alan Ashby	535.9259	545.1298	547.9009	541.0271	524.7372	541.7929	532.3366	535.4458	535.7940
-Alvin Davis	535.9259	528.8322	516.4257	519.6458	566.1561	620.0548	678.5273	677.8659	676.2552
-Andre Dawson	535.9259	591.8203	652.0612	686.7727	777.9026	809.6944	926.1300	939.7322	945.3525
-Andres Galarraga	535.9259	504.6060	467.6962	446.7049	376.4251	384.3876	455.4855	463.8098	467.5722
-Alfredo Griffin	535.9259	536.5405	545.4239	568.1979	614.8874	634.0209	614.9647	610.7495	606.2043
-Al Newman	535.9259	500.5316	460.2090	421.0248	296.2818	233.6435	164.0882	162.8150	162.4361
-Argenis Salazar	535.9259	503.6149	465.7149	439.6437	330.3456	276.3902	121.6240	106.5249	101.8118
-Andres Thomas	535.9259	503.2846	464.6474	441.0157	338.5642	288.2195	140.4565	134.7156	130.7999
-Andre Thornton	535.9259	597.5465	658.3527	677.6952	766.6450	780.7677	865.7657	867.6725	869.2311
-Alan Trammell	535.9259	555.0406	586.7019	614.6632	706.1221	740.4710	852.5354	857.0633	855.3116
-Alex Trevino	535.9259	520.0227	500.6597	475.0500	394.2121	365.2357	261.1921	257.6611	255.7363
-Andy VanSlyke	535.9259	522.0048	504.5612	499.7947	491.9494	480.3028	508.9806	515.6713	518.3513
-Alan Wiggins	535.9259	510.8828	493.4730	469.3028	383.5925	344.3632	314.6954	307.7798	305.8539
-Bill Almon	535.9259	531.4751	529.8499	507.0905	439.2594	402.6460	382.9395	385.2601	385.4946
-Buddy Bell	535.9259	608.8888	692.0716	740.8207	911.2037	966.3802	1022.8989	1032.7659	1035.5357

Figure 6.18: Lasso predictions

6.2.4 Elastic Net Regression

In this section the Elastic Net regression is going to be done, the package $elasticnet$ is needed, so one starts loading the library with the command **library(elasticnet)**, as in lasso regression the data must be on the same class, **X = as.matrix(Hitters[,-19])** and **y =Hitters[,19]**.

Using the next command the Elastic Net regression is done.

elastic < - enet(X,y,lambda = λ**)**

Depending on the value of λ one has a different elastic net regression, for $\lambda = 0$ the Lasso is obtained. One can observe that $Figure$ 6.16 and $Figure$ 6.19 are the same plot. This is obvious as elastic net is lasso when $\lambda = 0$. Another interesting case is when $\lambda = 1$, where the ridge regression is obtained. $Figure$ 6.20 displays the coeffi-cient paths of the elastic net when $\lambda = 1$.

The following table gives the MSE values, where the rows are the different values of the 8 first λ values, and the columns are the number of predictors are used in the model.

	0	1	2	3	4	5	6	7	...
0	202734	185851	169696	159221	131235	120530	104888	104370	...
0.125	202734	196696	168564	160599	149527	148439	135415	129337	...
0.25	202734	198748	173150	167465	161466	157560	151800	133154	...
0.375	202734	199615	179721	166218	162341	161371	159795	137547	...
0.5	202734	200094	183332	166462	164975	160421	159711	139536	...
0.625	202734	200397	185611	168382	163689	159528	159387	146523	...
0.75	202734	200607	187179	169848	162517	158882	158655	154065	...
0.875	202734	200760	188323	171008	161446	160049	158318	157797	...
1	202734	200877	189195	171950	162937	160393	157719	156939	...

Table 6.4: MSE for different values of λ on elastic net regression

6.3 Simulation study

In this section the data are simulate. We have 8 independent predictors and a response variable which is generated with the model:

$$\mathbf{y} = \mathbf{x}\beta + \epsilon, \qquad \epsilon \sim N(0, 10).$$

Where $\mathbf{x} \sim N(0, 1)$. The simulated data consists of a training set (50 values) observations and an independent test set (1000 values). β = (3, 1.5, 2, 0, 0, 0, 0, 0), so the predictors $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$ are significant for the model and the other predictors are noise.

6.3.1 Ordinary Least Square regression

On this section, an OLS regression will be done to the simulated database described in the previous section. The command to do the regression is:
ols <- **lm** (response ., data = Y)
After running **summary**(ols), the coefficients and some information are shown on R console are shown on$Figure$ 6.21

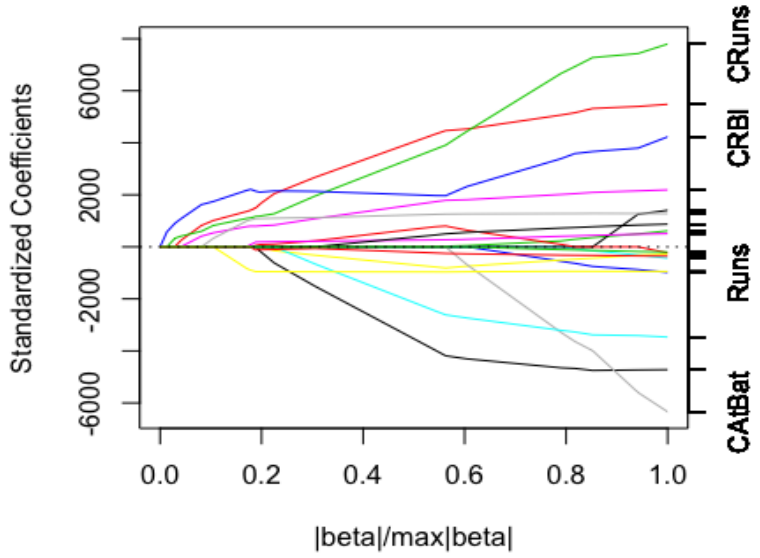

Figure 6.19: Elastic net regression. $\lambda = 0$

```
Call:
lm(formula = respuesta ~ ., data = Y_train)

Residuals:
     Min       1Q   Median       3Q      Max
-17.3020  -6.6697  -0.0387   6.9081  20.7632

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.2230     1.5001  -0.815  0.41962
x1            4.6945     1.7291   2.715  0.00965 **
x2            1.4842     1.5246   0.974  0.33601
x3            3.3076     1.2897   2.565  0.01408 *
x4            0.6176     1.6089   0.384  0.70305
x5            1.1823     1.4573   0.811  0.42186
x6            3.3680     1.5495   2.174  0.03556 *
x7           -0.2941     1.7879  -0.165  0.87014
x8            2.0450     1.9410   1.054  0.29824
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.3 on 41 degrees of freedom
Multiple R-squared:  0.3489,    Adjusted R-squared:  0.2218
F-statistic: 2.746 on 8 and 41 DF,  p-value: 0.01591
```

Figure 6.21: Summary of the OLS model

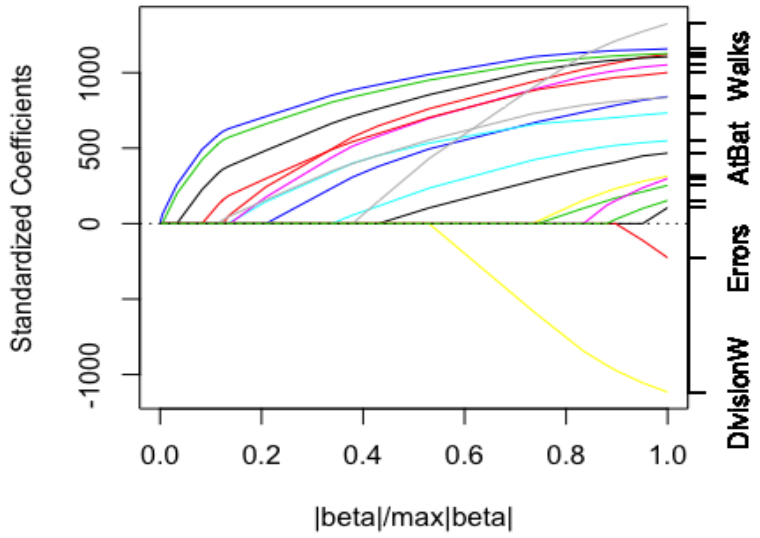

Figure 6.20: Elastic net regression. λ = 1

In $Figure$ 6.21, the $Pr(>|t|)$ column shows the p-values for the next hypothesis testing:

$$\begin{cases} H_0 : \beta_i = 0 \\ H_1 : \beta_i \neq 0 \end{cases} \tag{6.3}$$

Therefore, one can conclude that the variables for which the null hypothesis is rejected are $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_6$. Conversely, the variables for which the null hypothesis is plausible are the others variables, so those variables might not be significant on the OLS regression model, but it is wrong because as the model is generated, one knows that the variable $\mathbf{x}_2$ is significant and the variable $\mathbf{x}_6$ is not significant. So this model is not a good model. Moreover, observe that the R^2 coefficient is equal to 0.35, so the model may fit poorly the dataset. Using the test set, a prediction of the response is made in order to calculate the MSE, and it is obtained a MSE = 119.623.

6.3.2 Lasso regression model

In this section Lasso method is going to be used. At first one has to load the $lars$ package with the command **library(lars)**. Now, using the following command the Lasso regression is done. As the data must be in the correct class, **X1** is the train predictor matrix, and **y1** is the train response vector

lasso <-lars(X1,y1,type=c("lasso")),

with the following command the Lasso coefficient path is obtained.

plot(lasso)

$Figure$ 6.23 shows the coefficient path of the Lasso model. From the right to the left, one can see that the variable's coefficients are turning into zero in every algorithm step.

$Figure$ 6.24 shown the first 15 predictions obtained with Lasso regression. Observe that it is started on the observation 51, this is because the prediction is done on the test set. In the first column, the Lasso model does not have any variable, on the second column, the model has one variable, then successively.

$Figure$ 6.22 displays the coefficients of the method in all step of the algorithm. Observe that in every step one coefficient becomes a non-zero coefficient. Remark that for this method the $\mathbf{x}_6$, which is a noise variable, is more significant than $\mathbf{x}_2$, so it includes a noise variable in the model, one can discard the variables $\mathbf{x}_4$,$\mathbf{x}_5$ and $\mathbf{x}_7$ which are noise.

From the R-studio object lasso, one can extract the value of $\lambda = (\lambda_1, \ldots, \lambda_7)$, where λ_i is the value of λ on the step i. $\lambda_{intercept} \to \infty$ is the value for which all predictors are equal to 0 on the model. The following table is a three-column table. The first column displays the different values of λ. The second column displays the rations between λ_1 and the remaining values of λ. Finally the third column gives the mean squared error (MSE) of the predictions with the different values of λ.

λ_i values	$\frac{\lambda_i}{\lambda_1}$ proportions	MSE
$\lambda_{intercept} = \infty$	-	117.2976
λ_1 = 23.844675	1	116.7017
λ_2 = 22.524614	0.94463915	116.5146
λ_3 = 22.417098	0.94013018	107.5621
λ_4 = 10.469690	0.43907877	107.6049
λ_5 = 9.589658	0.40217188	111.8514
λ_6 =3.109187	0.13039337	113.8192
λ_7 = 2.077276	0.08711698	119.5968

Table 6.5: λ, proportions and MSE

Remark from 6.5 that the least MSE occurs when the model has 3 variables, this is when λ = 22.42, looking at $Figure$ 6.22 the model has the predictors $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_6$. So

with Lasso regression $\mathbf{x}_2$ is not significant. However, the MSE on the Lasso model is MSE = 107.5621 so this model has done a better prediction that OLS, but the predictor $\mathbf{x}_2$ still not being significant while it is.

```
              x1        x2        x3        x4        x5       x6        x7
[1,] 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
[2,] 0.00000000 0.0000000 0.1554936 0.0000000 0.0000000 0.000000 0.0000000
[3,] 0.02085794 0.0000000 0.1709989 0.0000000 0.0000000 0.000000 0.0000000
[4,] 2.53543266 0.0000000 2.0467207 0.0000000 0.0000000 1.957822 0.0000000
[5,] 2.71028902 0.0990644 2.1659962 0.0000000 0.0000000 2.108921 0.0000000
[6,] 4.20214388 0.8116782 3.0552423 0.0000000 0.9532602 3.114173 0.0000000
[7,] 4.43821100 0.9368140 3.2031940 0.2365324 1.1197821 3.332658 0.0000000
[8,] 4.96415689 1.2177897 3.5170334 0.6579332 1.5323993 3.802916 0.4668534
```

Figure 6.22: Lasso coeffcients

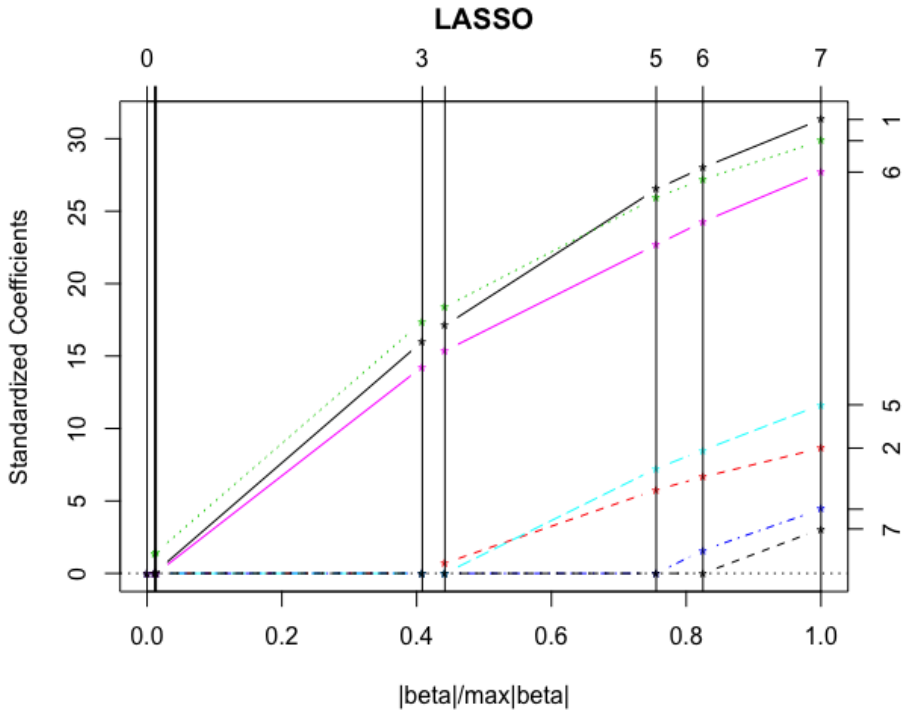

Figure 6.23: Lasso regression

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
51	-1.543741	-1.312097	-1.279565	1.304761489	1.59246423	4.492597591	4.862638723	5.04873732
52	-1.543741	-1.649577	-1.620780	-0.761430911	-0.70853341	0.761475697	0.984862140	1.01187806
53	-1.543741	-1.305480	-1.298364	-1.612157908	-1.69393055	-2.216427594	-2.104892381	-1.98848989
54	-1.543741	-1.706125	-1.698359	-4.144386516	-4.22480594	-3.380884010	-3.493392472	-3.38767505
55	-1.543741	-1.419934	-1.462656	-9.476665525	-10.15294086	-14.795505697	-15.204513851	-15.21833348
56	-1.543741	-1.446012	-1.435298	1.716130769	2.15800362	6.832065382	7.868064279	9.56252147
57	-1.543741	-1.762196	-1.744930	0.220809264	0.34333254	1.856421528	2.209438605	2.47207231
58	-1.543741	-1.660246	-1.648342	0.546298772	0.67209167	2.151821552	1.775472522	1.84286303
59	-1.543741	-1.629918	-1.609559	1.214433667	1.41505935	2.714580810	2.729594439	2.81498081
60	-1.543741	-1.270455	-1.226431	3.478484848	3.74937928	4.724095376	4.961978981	5.59777083
61	-1.543741	-1.626075	-1.623357	-3.577985736	-3.60119115	-4.213051363	-4.371512431	-5.59516524
62	-1.543741	-1.700398	-1.700820	-1.497177753	-1.61239244	-3.074404904	-3.594922201	-4.39056236
63	-1.543741	-1.633722	-1.620931	0.056719847	0.23694227	1.362764382	1.298026127	1.58560470
64	-1.543741	-1.664894	-1.669043	-1.683768059	-1.54509338	-2.296940699	-2.462506684	-3.18584200
65	-1.543741	-1.590905	-1.584469	0.046554314	-0.04923195	-1.680571484	-2.188399508	-3.11175134
66	-1.543741	-1.345236	-1.336132	0.005771419	-0.11870285	-1.556963114	-1.845347759	-2.16366455

Figure 6.24: Lasso predictions

6.3.3 Elastic Net Regression

In this section the Elastic Net regression is going to be done, the package $elasticnet$ is needed, so one starts loading the library with the command **library(elasticnet)**.

Using the next command the Lasso regression is done.

elasticsim < - enet(X1,y1,lambda = λ**)**

Depending on the value of λ one has a different elastic net regression, for $\lambda = 0$ the Lasso is obtained $Figure$ 6.25. $Figure$ 6.26 displays the coefficient paths of the elastic net when $\lambda = 1$.

The following table is the MSE table, where the rows are the different values of λ and the columns are the number of predictors are used in the model.
In this table, one can appreciate that the minimum value of the MSE is for $\lambda = 1$ and 4 predictors, this means that the best model is the ridge regression, as $\lambda = 1$, now the elastic net selects 4 variable. After seeing coefficients trace, the predictors which elastic net selects are $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_6$, so now, the three variables that generate the simulation are used are in the model. Moreover, the MSE is better than Lasso and OLS. MSE = 106.511.

	0	1	2	3	4	5	6	7
0	117.298	116.702	116.515	107.562	107.605	111.851	113.819	119.597
0.125	117.298	116.692	116.525	107.407	107.295	111.621	113.586	117.108
0.25	117.298	116.683	116.534	107.402	107.080	111.526	113.501	115.511
0.375	117.298	116.677	116.542	107.451	106.922	111.503	113.498	114.422
0.5	117.298	116.671	116.548	107.518	106.801	111.522	113.541	113.645
0.625	117.298	116.666	116.553	107.590	106.705	111.565	112.310	113.070
0.75	117.298	116.661	116.558	107.661	106.628	111.248	111.601	112.634
0.875	117.298	116.657	116.562	107.727	106.564	110.447	111.625	112.294
1	117.298	116.654	116.565	107.788	106.511	109.832	111.673	112.025

Table 6.6: MSE for different values of λ on elastic net regression

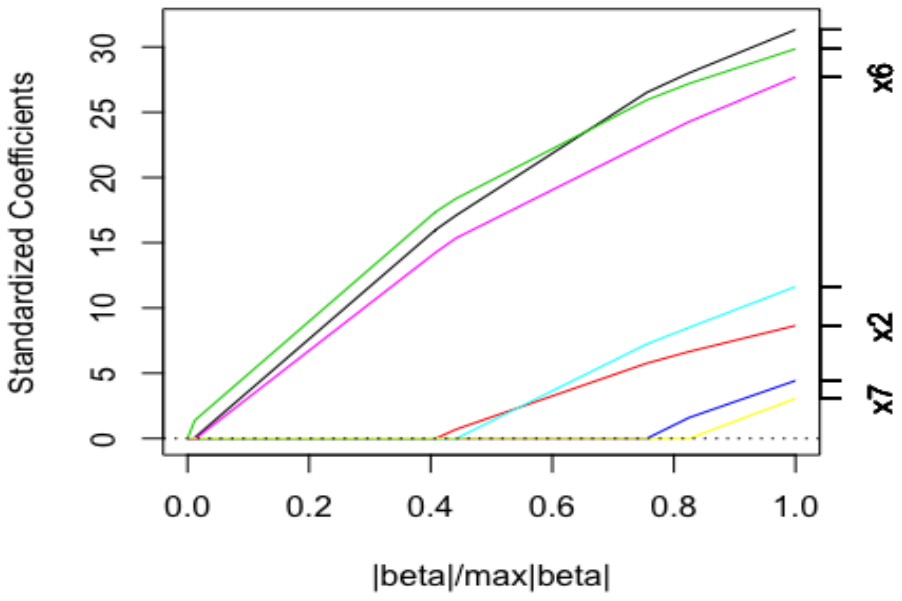

Figure 6.25: Elastic net regression. $\lambda = 0$

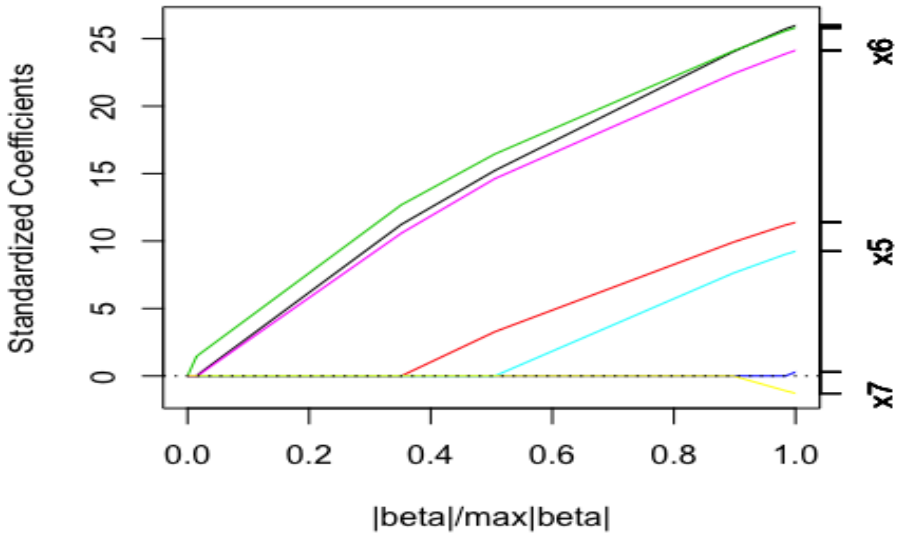

Figure 6.26: Elastic net regression. $\lambda = 1$

Bibliography

[1] ROBERT TIBSHIRANI, TREVOR HASTIE AND JEROME FRIEDMAN (2013) *Prostate database. url: https://web.stanford.edu/hastie/ElemStatLearn/*

[2] HUI ZOU AND TREVOR HASTIE. (2005) *Regularization and variable selection via the elastic net. Royal Statistical Society,* **67** (2):301–320

[3] JINZHU JIA AND BIN YU (2010) *On Model Selection Consistency of the Elastic Net when $p \gg n$. Statistica Sinica.*

[4] HOERL, A. AND KENNARD, R. (1988) *Ridge regression: Biased estimation for nonorthogonal problems. Technometrics.* **2**

[5] HÄRDLE, W.K. AND SIMAR, L (2019) *Applied Multivariate Statistical Analysis (Fifth Ed.). Springer.* **8**, **9** 251-283

[6] PETER BÜHLMANN AND SARA VAN DE GEER (2011) *Statistics for High-Dimensional Data Methods, Theory and Applications. Springer.* **4** 55-58

[7] B. EFRON, T. HASTIE, I JOHNSTONE, R. TIBSHIRANI (2004) *Least angle regression. Annals of statistics.* **2** 407 – 499

[8] P. ZHAO, B. YU (2006) *On model selection consistency of Lasso. The Journal of Machine Learning Research.* 2542-2549

[9] YUAN, M AND LIN, Y. (2007). *On the Nonnegative Garrote Estimator. Journal of the Royal Statistical Society* **69** 143-161

[10] G. JAMES, D. WITTEN, T. HASTIE AND R. TIBSHIRANI (2017) *Hiiters database:Data for an Introduction to Statistical Learning with Applications in R*

[11] T. HASTIE, R. TIBSHIRANI AND J. FRIEDMAN (2009) *The elements of statistical learning: Data Mining, Inference and Prediction. Springer Verlag* 61-79