Ca' Foscari
University
of Venice

Master's Degree programme
in **Computer Science**
Data Management and Analytics
Second Cycle (D.M. 270/2004)

Ca' Foscari
Dorsoduro 3246
30123 Venezia

Final Thesis

Multinomial Logistic Regression With High Dimensional Data

Supervisor
Ch. Prof. Cristiano Varin

Graduand
Rossana Salaro
Matriculation number 847168

Academic Year
2017/2018

Abstract

This thesis investigates multinomial logistic regression in presence of high-dimensional data. Multinomial logistic regression has been widely used to model categorical data in a variety of fields, including health, physical and social sciences. In this thesis we apply to multinomial logistic regression three different kind of dimensionality reduction techniques, namely ridge regression, lasso and principal components regression. These methods reduce the dimensions of the design matrix used to build the multinomial logistic regression model by selecting those explanatory variables that most affect the response variable. We carry out an extensive simulation study to compare and contrast the three reduction methods. Moreover, we illustrate the multinomial regression model on a case study that allows to highlight benefits and limits of the different approaches.

Contents

Introduction

The study of high-dimensional data has become in the last years one of the most important fields in computer sciences. Often the data that we are interested to study are characterized by a categorical response that might be multi-categorical. In this thesis we will study the multinomial logistic regression that aims to model data that are characterized by a multi-category response. In the study of high-dimensional data it is also useful to reduce dimensionality of the data. In this thesis we will describe some of the methods that perform reduction of dimensionality which are ridge regression, lasso and adaptive lasso, where the dimensionality reduction is done by shrinkage of the explanatory variables coefficients. We will also describe principal components regression for which dimensionality reduction is performed using linear combinations of the original explanatory variables, called principal components. Principal components are applied to the regression model as new explanatory variables. The different dimensionality reduction methods are then tested through a data simulation and finally applied to a real dataset containing description of the crimes reported by the New York City police department. The simulation considered different scenarios that test the characteristics and the limits of the dimensionality reduction methods applied to the multinomial logistic regression. Our analyses include also maximum likelihood estimation to compare against the standard fitting method that does not reduce the dimensionality of the problem.

Chapter 1

Multinomial Regression Models

This chapter defines regression models that relate explanatory variables to a multinomial response. This type of data arise frequently in applications. We can, for example, being interested into the forecasting of the kind of cancer that a certain patient suffers, given its RNA sequence. Different types of cancer may be influenced by different factors. Another example may be the analysis of food choices that alligators make. Adult alligators might have different preferences from young ones. Again, entering high school students make program choices among general program, vocational program and academic program. Their choice might be modelled using their writing score and their social economic status.
The rest of this chapter details logistic models for multinomial responses based on Agresti (2003).

1.1 Generalized Linear Models

Before to define what is the logistic regression model, we briefly recall the definition of generalized linear model and its main components. Generalized linear models extend ordinary regression models to encompass non-normal response distributions. A generalized linear model is defined by three components:

1. A *random component*, which identifies the response variable Y and its probability distribution;
2. A *systematic component*, which specifies the explanatory variables used to describe the mean response $E(Y)$;
3. A *link function*, which specifies the function of $E(Y)$ that the model equates to the systematic component.

The random component consists of a response variable Y with observations from a distribution in the natural exponential family, that has probability density function defined as

$$f(y_i, \theta_i) = a(\theta_i)b(y_i)\exp[y_i Q(\theta_i)]. \tag{1.1}$$

The value of the parameter θ_i may vary depending on values of explanatory variables. Quantity $Q(\theta_i)$ is known as *natural parameter*.
The systematic component, relates the linear predictor η_j to the explanatory variables using a linear model. Let x_{ij} be the value of explanatory variable j for subject i, with i=1,..., N and j = 1,..., p, then

$$\eta_i = \sum_j \beta_j x_{ij}.$$

The link function connects the random and the systematic components. Let $\mu_i = E(Y_i)$ denotes the expected response, $i = 1, ..., N$. The model connects μ_i with η_i by $\eta_i = g(\mu_i)$, where g is a monotonic differentiable function, namely we have

$$g(\mu_i) = \sum_j \beta_j x_{ij}, \qquad i = 1, ..., N.$$

1.1.1 Generalized Linear Models for Binary Data

We introduce the concept of generalized linear model for binary responses. This is the base structure of the model that we will going to construct and explain in the next sections. Let the response variable Y be binary. The possible outcomes of each observation are coded as 0 for failure and 1 for success. The expected value is $E(Y) = Pr(Y = 1|\, x)$. Then, denote Pr(Y=1) by $\pi(x)$, which reflects the dependence on the values of the explanatory variables.
A first option is to consider a *linear probability model* defined as

$$\pi(x) = \alpha + \beta x. \tag{1.2}$$

This specification correspond to a generalized linear model with binomial random component and identity link function, $\eta_i = \mu_i$. The linear probability model has an important drawback; probabilities fall in the range [0, 1], but linear functions of explanatory variables may take values over the entire real line. The linear probability model can be valid only over a restricted range of x values. When this is likely to happen, it is possible to interpret β as the amount of change in $\pi(x)$ for a one-unit increase in x.

1.2 Logistic Regression Models

Non-linear relationships between $\pi(x)$ and x are naturally considered with binary data. Often, a change in x has less impact when $\pi(x)$ approaches 0 or 1, rather than when this function is near to 0.5. This pattern is observed in the logistic regression model that assumes

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)}. \tag{1.3}$$

As x diverges, $\pi(x)$ monotonically approaches zero or one when $\beta < 0$ or $\beta > 0$, respectively. Figure (1.1) shows the typical S-shaped curves of the logistic regression model.

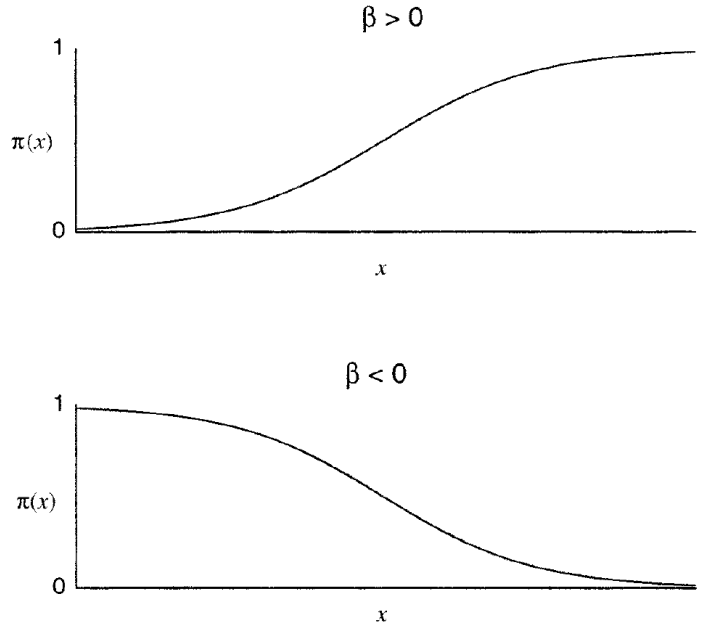

Figure 1.1: *Logistic regression function.*
Source: Agresti (2003). Categorical Data Analysis. 2nd Edition. Wiley Series in Probability and Statistics.

Logistic regression models are often expressed in terms of odds describing how more likely is a success than a failure. In the logistic model (1.3), the odds are defined as

$$\frac{\pi(x)}{1 - \pi(x)} = \exp(\alpha + \beta x).$$

Applying the logarithm (log-odds) to the previous formula we obtain the following linear relationship

$$\log\left[\frac{\pi(x)}{1-\pi(x)}\right] = \text{logit}[\pi(x)] = \alpha + \beta x. \tag{1.4}$$

Summarizing, logistic regression is a type of generalized linear model with binomial random component and a logit link function. For this reason logistic regression models are also known as *logit models*.
While, as said before, $\pi(x)$ needs to be in the range [0,1], the logit can assume any real value. Real numbers are also the range for linear explanatory variables that form the systematic component of a generalized linear model, then logit models do not suffer from the structural problem described before in the case of the linear probability model.

1.2.1 Interpretation of the Regression Parameters

The sign of β defines whether $\pi(x)$ increases or decreases when x increases. The rate at which the function increases or decreases depends on the size of $|\beta|$. The response variable Y is independent from X when $\beta = 0$. Given that the logistic density is symmetric the function $\pi(x)$ approaches 1 at the same rate that it approaches 0.
Exponentiating both sides of (1.4) we obtain (1.2).This shows that the odds are an exponential function of x. Hence, the odds increase multiplicatively by e^{β} for every 1-unit increase in x. In other terms, e^{β} is the odds-ratio given by the odds at $X = x + 1$ divided by the odds at $X = x$.

1.2.2 Multiple Logistic Regression

Like in the case of ordinary linear regression, multiple logistic regression, extends logistic models to multiple explanatory variables. Let $x = (x_1, ..., x_p)$ denote a generic vector of p explanatory variables. The model $\pi(x) = P(Y = 1|x)$ is defined as

$$\text{logit } \pi(x) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p, \tag{1.5}$$

that can be equivalently formulated as

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}. \tag{1.6}$$

The parameter β_i refers to the effect of x_i on the log odds that Y = 1, controlling the others x_j. Thus, at fixed levels of the others x_j, $\exp(\beta_i)$ is the multiplicative effect on the odds of a 1-unit increase in x_i while all the other explanatory variables x_j $(j \neq i)$ are kept fixed.

1.2.3 Likelihood-Ratio Test: Model Goodness of Fit

The Likelihood-ratio statistic $-2(L_0-L_1)$ tests whether all parameters added into the model M_1 are zero with respect to the model M_0. The models M_1 and M_0 must be nested models. The comparison is made between the log-likelihood L_1 for the fitted model M_1 with L_0 for the simpler model M_0. We denote this statistic with $G^2(M_0|M_1)$ for testing M_0, given that M_1 holds. There is also a special case in which the goodness-of-fit statistic $G^2(M)$ is defined with $M_0 = M$ and M_1 corresponds to the saturated model. To test if the model M is reasonable, we check if all parameters are zero within the saturated model, while in M they are not. The asymptotic degrees of freedom are defined by the difference of the number of parameters between the two models.
The likelihood-ratio statistic formula for comparing models M_1 and M_0 is

$$\begin{aligned} G^2(M_0|M_1) - 2(L_0 - L_1) \\ &= -2(L_0 - L_1) - [-2(L_1 - L_S)] \\ &= G^2(M_0) - G^2(M_1), \end{aligned}$$

where L_S denoted the maximized log-likelihood for the saturated model. In other words, the test statistic comparing two models, is identical to the difference of the goodness-of-fit statistics G^2 for the two models.

1.2.4 Fitting Logistic Regression Models

$$\pi(x_i) = \frac{\exp(\sum_{j=1}^{p} \beta_j x_{ij})}{1 + \exp(\sum_{j=1}^{p} \beta_j x_{ij})}. \tag{1.7}$$

Thereafter, it is convenient to incorporate the intercept in the vector of the coefficients β and consequently add an explanatory variable equal to 1. The standard fitting method for the logistic regression models is maximum likelihood. The likelihood function is

$$\begin{aligned} L(\beta) &= \prod_{i=1}^{N} \pi(x_i)^{y_i}[1 - \pi(x_i)]^{n_i - y_i} \\ &= \left\{ \prod_{i=1}^{N} \exp\left[\log\left(\frac{\pi(x_i)}{1 - \pi(x_i)} \right)^{y_i} \right] \right\} \left\{ \prod_{i=1}^{N} [1 - \pi(x_i)]^{n_i} \right\} \\ &= \left\{ \exp\left[\sum_i y_i \log \frac{\pi(x_i)}{1 - \pi(x_i)} \right] \right\} \left\{ \prod_{i=1}^{N} [1 - \pi(x_i)]^{n_i} \right\}. \end{aligned}$$

The i-th logit term is $\sum_j \beta_j x_{ij}$, then we have

$$L(\beta) = \Big\{ \exp[\sum_i y_i (\sum_j \beta_j x_{ij})] \Big\} \Big\{ \prod_{i=1}^{N} [1 - \pi(x_i)]^{n_i} \Big\},$$

that can be written as

$$L(\beta) = \Big\{ \exp[\sum_j (\sum_i y_i x_{ij}) \beta_j] \Big\} \Big\{ \prod_{i=1}^{N} [1 - \pi(x_i)]^{n_i} \Big\}.$$

Therefore, the log-likelihood is

$$l(\beta) = \sum_j \big(\sum_i y_i x_{ij} \big) \beta_j - \sum_i n_i \log \Big[1 + \exp \big(\sum_j \beta_j x_{ij} \big) \Big], \tag{1.8}$$

since $[1 - \pi(x_i)] = [1 + \exp(\sum_j \beta_j x_{ij})]^{-1}$. The log-likelihood depends on the binomial counts only through $\{\sum_i y_i x_{ij},\, j = 1, ..., p\}$, that is the sufficient statistic for β. The score function is

$$\frac{\partial l(\beta)}{\partial \beta_j} = \sum_i y_i x_{ij} - \sum_i n_i x_{ij} \frac{\exp(\sum_k \beta_k x_{ik})}{1 + \exp(\sum_k \beta_k x_{ik})},$$

and hence the likelihood equations are

$$\sum_i y_i x_{ij} - \sum_i n_i \hat{\pi}_i x_{ij} = 0, \qquad j = 1, ..., p, \tag{1.9}$$

where

$$\hat{\pi_i} = \frac{\exp(\sum_k \hat{\beta}_k x_{ik})}{1 + \exp(\sum_k \hat{\beta}_k x_{ik})},$$

is the maximum likelihood estimate of $\pi(x_i)$. The likelihood equations are not linear in β and require an iterative solution like the Newton-Raphson method.

1.3 The Multinomial Distribution

The multinomial distribution is the generalization of the binomial distribution to j possible outcomes. The multinomial distribution is characterized by a vector of success probabilities $p = (p_1, ..., p_j)$, where $p_i \geq 0$ and $\sum_{i=1}^{j} p_i = 1$.

Quantity p_i is the probability associated to the i-th outcome. Let X_i represents the number of times that the i-th outcome appears. We assume that $m = \sum_{i=1}^{j} X_i$. The joint probability function of $x = (x_1, ..., x_j)$ is defined as

$$Pr(X_1 = x_1, ..., X_J = x_J) = \binom{n}{x_1 \cdots x_J} p^{x_1} \cdots p^{x_J},$$

where

$$\binom{n}{x_1 \cdots x_J} = \frac{n!}{x_1! \cdots x_J!}.$$

Notice, the marginal distribution of X_i follows a binomial distribution with parameter p and size n.
The expected value and the variance of X are

$$E(X) = \begin{pmatrix} np_1 \\ \vdots \\ np_J \end{pmatrix},$$

$$\operatorname{Var}(X) = \begin{pmatrix} np_1(1-p_1) & -np_1p_2 & \cdots & -np_1p_J \\ -np_1p_2 & np_2(1-p_2) & \cdots & -np_2p_J \\ \vdots & \vdots & \vdots & \vdots \\ -np_1p_J & -np_2p_J & \cdots & np_k(1-p_J) \end{pmatrix}.$$

The maximum likelihood estimator of p is

$$\hat{p} = \begin{pmatrix} \hat{p_1} \\ \vdots \\ \hat{p_J} \end{pmatrix} = \begin{pmatrix} \frac{X_1}{n} \\ \vdots \\ \frac{X_J}{n} \end{pmatrix} = \frac{X}{n}.$$

The variance of the maximum likelihood estimator is

$$\operatorname{Var}(\hat{p}) = \operatorname{Var}(X/n) = n^{-2} \operatorname{Var}(X),$$

that is

$$\operatorname{Var}(\hat{p}) = \begin{pmatrix} p_1(1-p_1) & -p_1p_2 & \cdots & -p_1p_J \\ -p_1p_2 & p_2(1-p_2) & \cdots & -p_2p_J \\ \vdots & \vdots & \vdots & \vdots \\ -p_1p_J & -p_2p_J & \cdots & p_J(1-p_J) \end{pmatrix}.$$

1.4 Multinomial Logistic Regression

In section (1.2) we have considered the logistic regression model for binary categorical responses. In this section we will describe the multinomial logistic regression model, in which Y, the response variable, assumes J possible outcomes. Multi-category logit models for nominal response variables simultaneously describe log odds for all $\binom{J}{2}$ pairs of categories.
Let $\pi_j(x) = Pr(Y = 1|x)$ at x explanatory variables, where $\sum_j \pi_j(x) = 1$. We consider the joint counts of the J outcomes of Y as a multinomial random variable with probabilities $\{\pi_1(x), ..., \pi_J(x)\}$.
Logit models for multinomial responses are expressed in terms of a *baseline category*. This model simultaneously describes the effects of x on the J - 1 logits computed with respect to the baseline category, that is

$$\log \frac{\pi_j(x)}{\pi_J(x)} = \alpha_j + \beta_j' x, \qquad j = 1, ..., J-1, \tag{1.10}$$

assuming, without loss of generality, that the last category is the reference one. The described effects vary with respect to the baseline category paired with the response. These J - 1 equations determine the parameters for the logits with other pairs of response categories since

$$\log \frac{\pi_a(x)}{\pi_b(x)} = \log \frac{\pi_a(x)}{\pi_J(x)} - \log \frac{\pi_b(x)}{\pi_J(x)}.$$

Therefore, the probability of the j-th outcome is

$$\pi_j(x) = \frac{\exp(\alpha_j + \beta_j' x)}{1 + \sum_{h=1}^{J-1} \exp(\alpha_h + \beta_h' x)}, \quad j = 1, ..., J, \tag{1.11}$$

where $\alpha_J = 0$ and $\beta_J = 0$ for model identifiability.

1.4.1 Fitting of Baseline-Category Logit Models

Let $y_i = (y_{i1}, ..., y_{iJ})$ represents the multinomial trial for subject i, with $y_{ij} = 1$ when the response is in category j and $y_{ij} = 0$ otherwise. Thus we have, $\sum_j Y_{ij} = 1$. Moreover $x_i = (x_{i1}, ..., x_{ip})'$ is the vector of the explanatory variables values for subject i and $\beta_j = (\beta_{j_1}, ..., \beta_{jp})'$ is the parameters vector for the j-th logit.

The contribution to the log-likelihood by subject i is

$$\log\left[\prod_{j=1}^{J}\pi_j(x_i)^{y_{ij}}\right] = \sum_{j=1}^{J-1} y_{ij}\log\pi_j(x_i) + \left(1-\sum_{j=1}^{J-1}y_{ij}\right)\log\left[1-\sum_{j=1}^{J-1}\pi_j(x_i)\right]$$
$$= \sum_{j=1}^{J-1} y_{ij}\log\frac{\pi_j(x_i)}{1-\sum_{j=1}^{J-1}\pi_j(x_i)} + \log\left[1-\sum_{j=1}^{J-1}\pi_j(x_i)\right],$$

since $\pi_J = 1-(\pi_1+\cdots+\pi_{J-1})$ and $Y_{iJ} = 1-(y_{i1}+\cdots+y_{i,J-1})$. Assume N independent observations, then the log likelihood is

$$\log\prod_{i=1}^{n}\left[\pi_j(x_i)^{y_{ij}}\right]$$
$$= \sum_{i=1}^{n}\left\{\sum_{j=1}^{J-1} y_{ij}(\alpha_j+\beta_j'x_i) - \log\left[1+\sum_{j=1}^{J-1}\exp(\alpha_j+\beta_j'x_i)\right]\right\}$$
$$= \sum_{j=1}^{J-1}\left[\alpha_j\left(\sum_{i=1}^{n}y_{ij}\right)+\sum_{k=1}^{p}\beta_{jk}\left(\sum_{i=1}^{n}x_{ik}y_{ij}\right)\right].$$

The sufficient statistic for β_{jk} is $\sum_i x_{ik}y_{ij}$, $j=1,...,J-1$ and $k=1,...,p$. With respect to α_j the sufficient statistic is $\sum_j y_{ij}$.
The likelihood equations equate the sufficient statistics to their expected values. The log likelihood equation is concave and to obtain the maximum likelihood estimates of β we use the iterative Newton-Raphson method. The procedure is based on an initial guess on the solution. It then obtains a sequence of guesses that are computed as approximations of the function to maximize in a neighbourhood of the previously obtained guess. The resulting estimates have large-sample normal distribution and their asymptotic standard errors are the square roots of the diagonal elements of the inverse Fisher information matrix. An alternative fitting method consists in estimating logit models separately for the J-1 pairing of responses. A logit model fitted using this approach is the same as a regular logit model fitted conditionally on the classification into one of the categories. The j-th baseline-category logit assumes conditional probabilities

$$\log\frac{\pi_j(x)/[\pi_j(x)+\pi_J(x)]}{\pi_J(x)/[\pi_j(x)+\pi_J(x)]} = \log\frac{\pi_j(x)}{\pi_J(x)}.$$

The separate-fitting estimates differ from the maximum likelihood estimates based on simultaneous fitting of the J-1 logits. The separate-fitting approach losses efficiency because it tends to have larger standard errors. However, the loss in efficiency is minor if the response category having the highest prevalence is the baseline category.

Chapter 2

Dimensionality Reduction Methods

In this chapter we will focus on those methods that allow us to reduce the dimensionality of the design matrix $\mathbf{X}$. Firstly we will discuss about *ridge regression* and the *lasso*, which shrink the regression coefficients using a penalty term. The two methods act in similar ways but differ in the nature of the penalty term and so in the complexity of the solution. We start with the presentation of ridge and lasso separately and after that we will compare the two methods to understand their similarities and differences. We will also describe the *adaptive lasso* that assigns a weighted penalization to the estimated coefficients.
The last technique that we will consider is the *principal components regression* method, which is derived from principal components analysis. This method is quite different from the first two because it uses the linear combination of the design matrix $\mathbf{X}$ within the regression model. This set of linear combinations allows for dimensionality reduction. We will discuss also some similarities between the principal components regression method and the ridge regression method.
The theoretical notions of the next sections came from Hastie et al. (2008) for the ridge and the lasso regression methods, and from Hastie et al. (2008) when we describe the principal components regression.

2.1 Ridge Regression

Before to introduce the method of ridge regression, let us recall some basic notions of linear models and least squares. Given a a design matrix $X^T =$

$(X_1, X_2, ..., X_p)$, we want to predict the response Y through the linear model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j,$$

that can also be expressed in matrix form as

$$\hat{Y} = \hat{\beta}_0 + X^T \hat{\beta},$$

where, $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$. The mos popular fitting method for linear models is *least squares* that selects the coefficient β to minimize the residual sum of squares

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2.$$

The residual sum of squares is a quadratic function of the parameters, its minimum always exists, but it may not be unique. The solution to this problem can be expressed in matrix form. Write

$$\mathrm{RSS}(\beta) = (y - X\beta)^T (y - X\beta),$$

where $\mathbf{X}$ is a $N \times p$ matrix corresponding to the explanatory variables, and $\mathbf{y}$ is an N-vector of the outputs in the training set. Differentiating with respect to β, we get the normal equations

$$X^T (y - X\beta) = 0.$$

When $X^T X$ is non-singular, the *unique solution* of the problem is given by

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

The fitted value of the i-th input is given by $\hat{y}_i = \hat{y}(x_i) = x_i^T \hat{\beta}$.
After this short recall about linear regression methods and least squares, we can introduce the concept of ridge regression that is part of the shrinkage methods. Ridge regression shrinks the coefficient of the regression by imposing a *penalty* on their size. The ridge coefficients minimize a *penalized residual sum of squares* so that

$$\hat{\beta}_{ridge} = \underset{\beta}{\mathrm{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \qquad (2.1)$$

where $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ the greater the shrinkage. Ridge regression can also be expressed as

$$\hat{\beta}_{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij\beta_j}\right)^2,$$
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t. \tag{2.2}$$

The parameters λ and t in the formulas (2.1) and (2.2) are in one-to-one correspondence. Formulation (2.2) makes clear the shrinking effect on the parameters size.
One motivation for ridge regression arises with correlated explanatory variables. When there are many correlated variables in the linear regression model, their coefficients can be poorly determined and may exhibit high variance. Indeed, a widely large positive coefficient on one variable, can be deleted by a similarly large negative coefficient on its correlated cousin. This phenomenon is alleviated imposing a size constraint on the coefficients . It is important to notice that ridge solutions are not equivariant under scaling of explanatory variables, and for this reason it is advisable to standardize the explanatory variables before computing the ridge solutions.
Another important aspect to take into account is that the intercept has been left out of the penalty term. The reason of this choice is that including the intercept into the penalization term will make the procedure dependent on the origin chosen for the response variable Y. This choice will add a constant to each of the target y_i but will not result in a simple shift of the predictions by the same amount. Therefore it is convenient to derive the solution of (2.1) centring the explanatory variables. Proceeding in this way, each x_{ij} is replaced by $x_{ij} - \bar{x}_j$. The intercept is then estimated as $\bar{y} = \frac{1}{N}\sum_1^N y_i$. The penalized residual sum of squares previously expressed in (2.1) can also be rewritten in matrix form as

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

From this statement, the ridge regression solutions are

$$\hat{\beta}_{ridge} = (X^TX + \lambda I)^{-1}X^Ty, \tag{2.4}$$

where I is the $p \times p$ identity matrix. With the quadratic penalty $\beta^T\beta$, the solution of the ridge regression is a linear function of the response variable $\mathbf{y}$. The ridge solution adds a constant to the diagonal of X^TX before inversion.

This transformation makes the problem non-singular even in case in which the matrix $X^T X$ is not of full rank.
In case of orthonormal explanatory variables, the ridge regression estimates are just a *scaled* version of the least squared estimates $\hat{\beta}_{ridge} = \hat{\beta}/(1+\lambda)$.
If we want to look more into the details of the ridge regression nature, we can analyse the singular value decomposition of the centred design matrix $\mathbf{X}$ of input defined as

$$\mathrm{X} = \mathrm{UDV}^T. \tag{2.5}$$

In this equation, the matrix $\mathbf{U}$ is the $N \times p$ orthogonal matrix that spans the column space of $\mathbf{X}$, while the matrix $\mathbf{V}$ is the $p \times p$ matrix that spans the row space of $\mathbf{X}$. The matrix $\mathbf{D}$ is diagonal with entries $d_1 \geq d_2 \geq ... \geq d_p \geq 0$ called singular values of $\mathbf{X}$. The design matrix is *singular* when at least one d_j is null. Using the single value decomposition we can express the least squares fitted vector as

$$\begin{aligned} y_{ols} = X\hat{\beta}_{ols} &= X(X^T X)^{-1} X^T y \\ &= UU^T y. \end{aligned} \tag{2.6}$$

Accordingly, the fitted values in ridge regression are

$$\begin{aligned} y_{ridge} = X\hat{\beta}_{ridge} &= X(X^T X + \lambda I)^{-1} X^T y \\ &= UD(D^2 + \lambda I)^{-1} DU^T y \\ &= \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y, \end{aligned} \tag{2.7}$$

where the u_j are the columns of $\mathbf{U}$. Since $\lambda \geq 0$, we have that $d_j^2/(d_j^2 + \lambda) \leq 1$. Ridge regression computes the coordinates of $\mathbf{y}$ with respect to the orthonormal basis $\mathbf{U}$, then it shrinks these coordinates by $d_j^2/(d_j^2 + \lambda)$. The single value decomposition of the centred matrix $\mathbf{X}$ gives the expression of the *principal components* of $\mathbf{X}$. The sample covariance is given by $S = X^T X/N$ and the eigen decomposition of $X^T X$ is

$$X^T X = V D^2 V^T.$$

The columns of the matrix $\mathbf{V}$ correspond to the eigenvectors v_j and are the principal components directions of $\mathbf{X}$. For the first principal component direction v_1, we have that $z_1 = Xv_1$, that has the largest sample variance amongst all normalized linear combinations of the columns of $\mathbf{X}$. The sample variance of z_1 is

$$Var(z_1) = Var(Xv_1) = \frac{d_1^2}{N}. \tag{2.8}$$

The vector z_1 is called the principal component of $\mathbf{X}$ and u_1 is the normalized first principal component. The last principal component has minimum variance, hence the small singular values d_j correspond to the directions in the column space of $\mathbf{X}$ having small variance, and then ridge regression shrinks these directions the most. We will encounter this concept later on, analysing the principal component regression.
The implicit assumption of ridge regression is that the response will vary the most in the direction of high variance of the inputs. The largest principal component is the direction that maximizes the variance and the smallest principal component is the direction that minimizes the variance. Ridge regression projects $\mathbf{y}$ onto these components and then shrinks the coefficients of the low-variance components more than those with high-variance.

2.2 The Lasso

Lasso is a shrinkage method like ridge regression, but with substantial differences. Lasso estimator is defined as

$$\begin{aligned} \hat{\beta}_{lasso} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \\ &\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t. \end{aligned} \tag{2.9}$$

As in the case of ridge regression the intercept is not penalized. Again it is advisable to standardize the explanatory variables in such a way that the estimator of the intercept is $\bar{y}$.
The equivalent, *Lagrangian form* of lasso is given by

$$\hat{\beta}_{lasso} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{2.10}$$

This formula is very similar to the one of ridge regression, with the important difference that the penalty term is now defined as $\sum_1^p |\beta_j|$. The latter makes the solution non-linear in the responses y_i. The solution of lasso can be computed by solving a *quadratic programming problem*. Efficient algorithms are available to solve the entire path of solutions for lasso with the same computational cost needed for ridge regression. The path of solution for lasso is obtained repeating the computation $\hat{\beta}$, changing the value assumed by the shrinkage term λ.

Because of the nature of the constraint used in lasso, making t sufficiently small will lead some of the coefficients to be *exactly* equal to zero. The parameter t should then be adaptively chosen to minimize an estimate of expected prediction error, like in the case of the choice of the penalty term in ridge regression.

2.3 The Adaptive Lasso

What we have seen in (2.10) as Zou (2006) says, shows that lasso penalizes all coefficients of the same amount independently of their size. Instead, the weighted lasso Zou (2006) assigns distinct

$$\beta_{wlasso} = \underset{\beta}{argmin} \left|\left| y - \sum_{j=1}^{P} x_j \beta_j \right|\right|^2 + \lambda \sum_{j=1}^{P} w_j |\beta_j|$$

where the w_j are known weights. If the weights are data-dependent and properly chosen, then the weighted lasso can have the oracle properties and it is known as *adaptive lasso*. The oracle property is that the asymptotic distribution of the estimator is the same as the asymptotic distribution of the maximum likelihood estimator computed on only the true support, that is the subset of x whose true coefficients are not null. That is the estimator adapts to knowing the true support without paying a price in terms of the asymptotic distribution (Zou, 2006).
Adaptive lasso is a regularization method that avoids over-fitting, penalizing coefficients. It has the same advantage of lasso: it can shrink some of the coefficients exactly to zero, performing thus a selection of the attributes with the regularization. Adaptive lasso seeks to minimize

$$RSS(\beta) + \lambda \sum_{j=1}^{P} \hat{w}_j |\beta_j|,$$

where λ is the tuning parameter typically chosen through cross validation. Weights $\hat{w}_j$ perform a different regularization on each coefficient. Zou (2006) proposes to set $\hat{w}_j$ to

$$\hat{w}_j = \frac{1}{\left(|\hat{\beta}_j^{ini}|\right)^{\gamma}},$$

where $\hat{\beta}_j^{ini}$ is an initial estimate of the coefficients, usually obtained through ridge regression. Adaptive lasso ends up penalizing those coefficients with lower initial estimate. The parameter γ is a positive constant for adjustment of the adaptive weights vector and can be fixed at 0.5, 1, or 2.

2.4 Comparison between Ridge Regression and the Lasso

In the case of orthonormal design matrix $\mathbf{X}$, ridge and lasso have an explicit solution. Each of them applies a simple transformation to the least squares estimates $\hat{\beta}_j$ that is summarized in Table 2.1. In Table 2.1, where λ is the

Estimator	Formula
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\hat{\beta}_j - \lambda)_+$

Table 2.1: *Ridge and lasso estimator for the orthonormal design matrix.*

shrinkage constant, *sign* denotes the sign function and x_+ means the positive part of x.

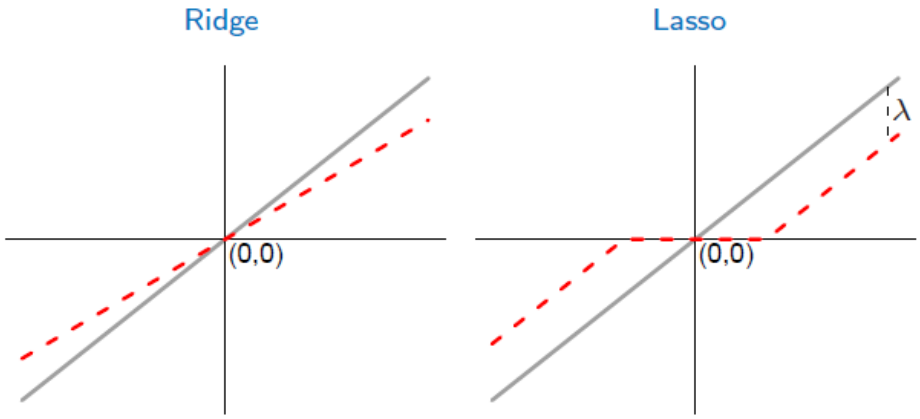

Figure 2.1: *Ridge and lasso estimators (dashed red lines) compared to the ordinary least squares estimate (solid grey line). Source: Hastie et al. (2008). The Elements of Statistical Learning, Data Mining, Inference, and Prediction. 2nd Edition. Springer Series in Statistics.*

Ridge regression does a *proportional shrinkage*, while lasso translates each coefficient by a *constant factor* λ, truncating at zero: this procedure is known as *soft thresholding*. The fact that lasso truncates at zero some of the coefficient is illustrated in Figure 2.1.
To analyse the non-orthogonal case, we will describe ridge and lasso with the

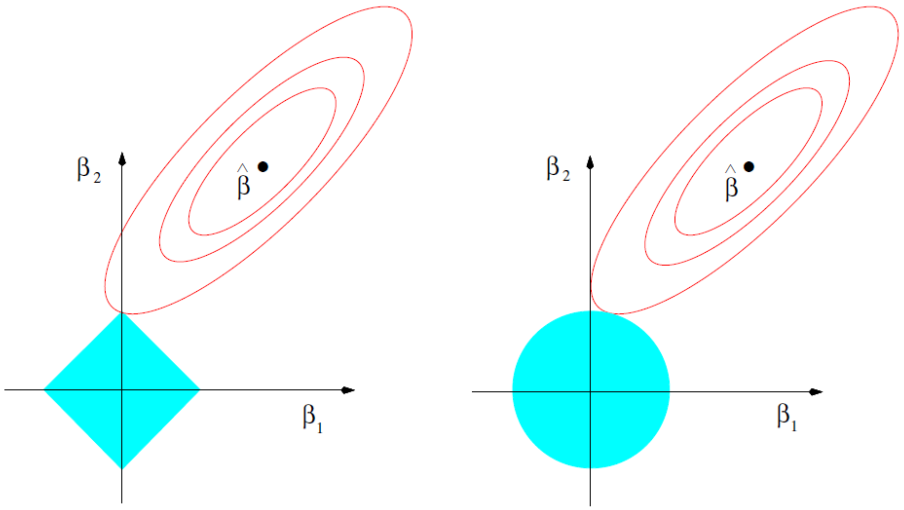

Figure 2.2: *Estimation for the lasso (left) and ridge (right). Source: Hastie et al. (2008). The Elements of Statistical Learning, Data Mining, Inference, and Prediction. 2nd Edition. Springer Series in Statistics.*

help of some images about two parameters.
Figure 2.2 shows the residual sum of squares that has elliptical contours, in the figure plotted in red, centred at the full least squares estimate. The constraint region for ridge regression is the disk $\beta_1^2 + \beta_2^2 \leq t^2$, while the diamond represents lasso constraint region, that is given by $|\beta_1| + |\beta_2| \leq t$. Both approaches find the first point where the elliptical contours hit the constraint region. In particular, looking at the diamond, if the solution occurs at the corner, this means that there is a parameter $\beta_j = 0$. When $p > 2$, there are many more chances for the estimated parameters to be equal to zero.
It is also possible to generalize ridge and lasso, considering them as maximum a posteriori estimators of type

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}, \tag{2.11}$$

for some $q \geq 0$. Thinking of $|\beta_j|^q$ as the log-prior density for β_j, the $|\beta_j|^q$ are also the equi-contours of the prior distribution of the parameters. In particular, we have that when q=1, $|\beta_j|$ correspond to the lasso estimators and when q=2, $|\beta_j|^2$ are the ridge regression estimators. When $q \leq 1$, the prior is not uniform in direction, but it concentrates more mass in the coordinates directions. For q=1 we have the smallest value such that the constraint region is convex. If we are considering a non-convex region we can face more

difficulties is solving the optimization problem.
Ridge and lasso are derived as posterior modes, but it is more common to use the posterior mean. Ridge estimator it is also a posterior mean, but this is not true for lasso.
The *elastic-net* penalty is a compromise between the lasso and ridge regression

$$\lambda \sum_{j=1}^{p} (\alpha \beta_j^2 + (1-\alpha)|\beta_j|),$$

where α determines the combination of ridge and lasso penalty.

2.5 Principal Component Regression

The description of *principal components regression* requires some basic notions of principal component analysis. Principal components are linear combinations of the explanatory variables, ordered by an informative criterion. The first principal component extracts from data the maximum quantity of information; the next principal components optimize the residual information, under the constraint to be uncorrelated with the other components. In general, if we have observed $p>1$ variables $X_1, ..., X_p$ we can determine the same amount of linearly independent principal components $Z_1, ..., Z_p$. From this set of linearly independent principal components we consider the subset of q principal components $(1 < q < p)$ that explains a significant portion of the information, or typically the variance, contained within the explanatory variables.
Principal components are often used to obtain dimensionality reduction. Indeed, the projection of the data into the space of the first q principal components allow us to study the statistical properties within a smaller space, hopefully with a limited loss of information. Principal components, as we will se later on, are also used with regression models when the explanatory variables are strongly correlated. In this particular case the parameters estimates are unstable or they do not exist. Then, a possible approach to solve this problem is to substitute the explanatory variables $X_1, ..., X_p$ with the correspondent principal components $Z_1, ..., Z_p$, or a subset of them.
The main problem of principal components is their interpretation because they are artificial variables, obtained as a linear combination of the original input variables. Principal components are typically computed after standardization of the design matrix. In case of standardization, the covariance matrix of the standardized data coincides with the correlation matrix of the explanatory variables and for this reason the principal components will be

based on the eigenvalues and eigenvectors of the correlation matrix. The number of principal components is chosen imposing a threshold on the explained variability by the principal components.
In many situations we need to take into consideration a large number of explanatory variables and often they are strongly correlated. Through the use of principal component we can obtain a small number of linear combinations of the original explanatory variables X. The Z are then used in place of the X as input in the regression procedure.
Principal component regression is a two-step method. First it forms the derived input columns $z_m = Xv_m$, where v_m are the principal components directions as we have seen in the case of ridge regression. The second step is regressing y on $z_1, z_2, ..., z_M$ for some $M \leq p$. Since the z_m are orthogonal, this regression is simply a sum of univariate regressions

$$\hat{y}^{(M)}_{pcr} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_m, \tag{2.12}$$

where $\hat{\theta} = \langle z_m, y\rangle / \langle z_m, z_m\rangle$ is the ratio between the scalar product of the vector z_m and y and the scalar product of z_m and itself. Since the z_m are linear combinations of the original x_j we can express the equation we have defined in (2.12), in terms of coefficients of the x_j

$$\hat{\beta}^{(M)}_{pcr} = \sum_{m=1}^{M} \hat{\theta}_m v_m. \tag{2.13}$$

As for ridge regression, principal components depends on the scale of the explanatory variable, for this reason the analysis if performed after standardization. Note that if M=p, we would just get back to the usual least squares estimates since the columns of $\mathbf{Z}$=$\mathbf{UD}$ span the column space of $\mathbf{X}$. When M<p, we get a reduced regression. We can understand that principal components regression is very similar to ridge regression: indeed they both operate via the principal components of the design matrix. The difference is that with ridge regression we shrink the coefficient of the principal components, the degree of shrinkage depending on the size of the corresponding eigenvalue, while in principal components regression we discard the $p - M$ smallest eigenvalue components. Figure 2.3 illustrate the difference between ridge regression and principal components regression.

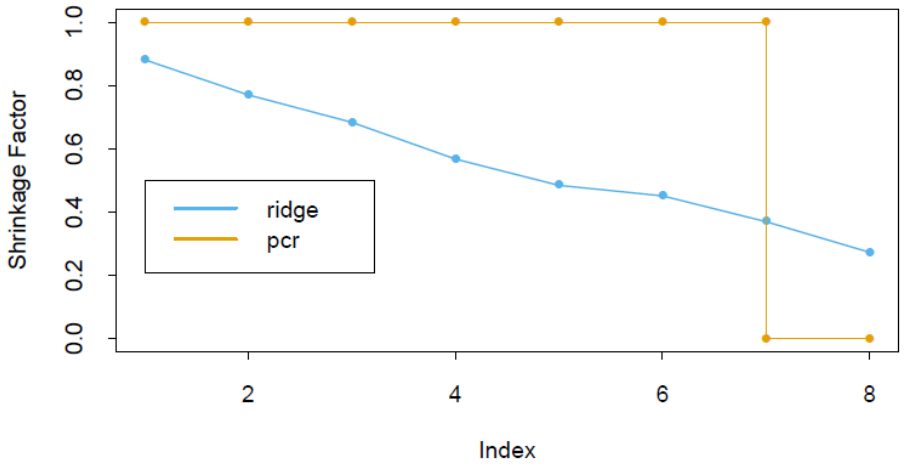

Figure 2.3: *Ridge regression shrinks the regression coefficients of the principal components using the shrinkage factor* $d_j^2/(d_j^2+\lambda)$*, while principal component regression truncates them. Source: Hastie et al. (2008). The Elements of Statistical Learning, Data Mining, Inference, and Prediction. 2nd Edition. Springer Series in Statistics.*

Chapter 3

Simulation study

This chapter illustrates a simulation study that aims to outline the characteristics of the methods described in Chapter 2. After the illustration of the structure of this simulation we will provide the results with respect to the log-score of the estimated probabilities and the prediction accuracy of each method. Then, we will present the distributions of the estimated coefficients with the different methods. We will also give a measure of accuracy about the coefficients selection performed by the two lasso methods considered. Finally, we will compare the methods in terms of root mean square error and we will describe the principal components selected for principal components regression.Simulations and the real data application are developed using the R programming language (R Core Team, 2018).). Computations are made with the R package `glmnet` (Friedman et al., 2010) that fits multinomial regression models using their Poisson log-linear representation (Rodríguez, 2007).

3.1 Description of the Simulation

The simulation takes into account a categorical response variable with three levels simulated from the multinomial logistic regression model described in Chapter 1. The design matrix $\mathbf{X}$ is simulated from a multivariate normal distribution with zero mean, unit variance and exponential correlation between the explanatory variables of type

$$Cor(x_i, x_j) = \rho^{|i-j|}.$$

The correlation parameter ρ was set equal to 0.0, 0.3, 0.6 and 0.9 in order to represents different degrees of association between the explanatory variables.

In the simulation study we consider a designed matrix with 20 simulated explanatory variables. We considered four different scenarios that differ in how much the explanatory variables of the design matrix are correlated and where the sample dimension is $n = 500$. We performed an additional scenario with $n = 200$ observations and correlation parameter equals to 0.6. We choose to analyse the case with 200 observations to evaluate the behaviour of the regression fitting methods when the amount of information available for each coefficient of the model is limited. All the five scenarios were replicated 500 times.
The coefficients β and the intercepts α of the multinomial model are defined as:

$$\beta = (3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$\alpha = (-0.5, 0.5, 0.0).$$

The values of β have been chosen to reflect an hypothetical scenario that includes important ($\beta = 3$), moderately relevant ($\beta = 1$) and irrelevant ($\beta = 0$) explanatory variables. These two vectors are used to obtain the probability matrix. The probabilities of the model have been computed according to (1.11) and are used to generate the multinomial response vector Y. In this simulation the baseline category corresponds to the last category.
Once the design matrix X and the response vector Y have been simulated we performed the multinomial logistic regression using five different methods:

1. Maximum likelihood;
2. Ridge regression;
3. Lasso;
4. Adaptive lasso;
5. Principal component regression.

We split the design matrix $\mathbf{X}$ and the response vector $\mathbf{Y}$ into *train* and *test* sets. The train set consider the 80% of the data, while the test set the remaining 20% of the data.
The penalty parameter of ridge and lasso regression was selected with ten-fold cross-validation. The measure used for model selection with cross-validation is the *miss-classification error*. We computed the predictions on the test set using the best λ obtained from the cross-validated model which is the lambda that is at one-standard error of distance from the minimum of the cross-validated miss-classification error (Hastie et al., 2008).

We evaluated and compared the five methods through the log-score defined as

$$L = -\sum_{i \in test} \log\left[\hat{Pr}(Y_i = y_i|x_i)\right],$$

where $\hat{Pr}$ denotes the estimated prediction probability using the information available in the train test. We provided the prediction accuracy percentage given by the number correct predictions.
Thereafter, we illustrate the estimated coefficients with three representative coefficients, namely:

1. Coefficient $\beta_1 = 3$ that has an high effect on the model response;
2. Coefficient $\beta_9 = 1$ that has a moderate effect on the model response;
3. Coefficient $\beta_{16} = 0$ that has a negligible effect on the model response.

The distribution of the estimated coefficients are represented using multiple paired-boxplots. For the two lasso methods we describe the percentage of correctly selected explanatory variables. For the maximum likelihood, the ridge and the two lasso methods we also report the *root mean square error* defined as:

$$RMSE = \sqrt{E(\hat{\beta}_j - \beta_j)^2 + Var(\hat{\beta}_j)}.$$

With respect to principal components regression, we reported the number of principal components selected in a way to reach 70% of the overall standard deviation.

3.2 Simulations Results

3.2.1 Log-Score

Figure 3.1 refers to the scenario with the correlation parameter equal to zero. The figure shows that the model with the minimum log-score is the maximum likelihood, followed by adaptive lasso and lasso. Principal components regression and ridge show the higher values of the log-score in this scenario. Ridge regression is the method with the highest log-score. Lasso and adaptive lasso log-scores are more or less equivalent. The two lasso methods have the less variable log-scores, while principal components regression, ridge and maximum likelihood fitting methods have more variable log-scores.

Figures 3.2 and 3.3 display the log-scores when the correlation parameter is 0.3 and 0.6, respectively. The log-scores of maximum likelihood have an

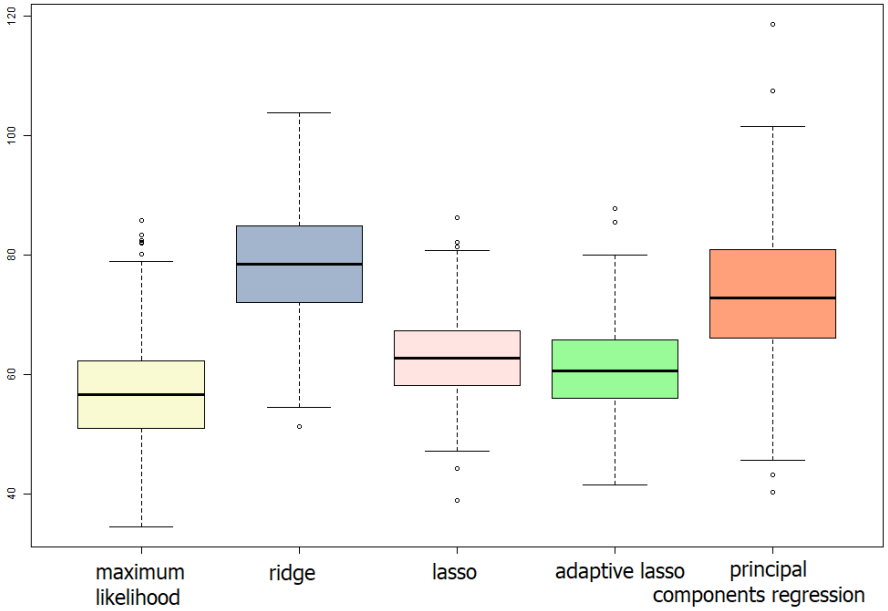

Figure 3.1: *Boxplots of the simulated log-scores for the fitting methods when the correlation parameter is equal to 0.*

high variability. Ridge is the method with the highest log-score. Lasso and adaptive lasso seem to suffer a little bit more the variability of the predictions, and their log-scores are quite similar. The log-score of principal components regression improves with respect to the case of the correlation parameter equal to zero. As the correlation between the explanatory variables increases the number of grossly incorrect predictions in all the multinomial regression fitting methods rises. Figure 3.3 shows that maximum likelihood and ridge have the highest variability in the log-scores. Lasso and adaptive lasso log-scores tend to be more concentrated. They maintain, as in the case of correlation equal to 0.3, a relatively low variability in log-scores compared with the other three methods. Principal components regression has the best log-score among the considered methods when the correlation parameter is 0.6.

Figure 3.4 reports the result of the scenario with correlation parameter equal to 0.9. Maximum likelihood has an high variability in log-scores. We observe some grossly incorrect predictions in all the regression fitting methods. Ridge maintains its log-score similar to those presented in the previous scenarios. The variability in log-scores of ridge in this case is a little bit lower

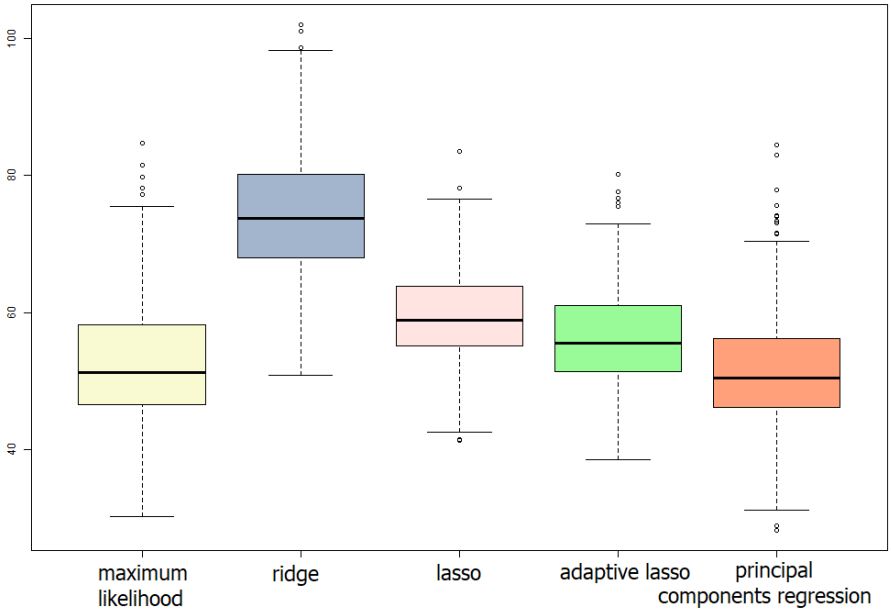

Figure 3.2: *Boxplots of the simulated log-scores for the fitting methods when the correlation parameter is equal to 0.3.*

than lasso and adaptive lasso. Lasso and adaptive lasso log-scores increased and also their variability. Principal components regression is the method with the lower log-score. Principal components regression variability is more or less similar to the one we observe in lasso and adaptive lasso.

Figure 3.5 provides the scenario with correlation parameter equal to 0.6 and $n = 200$ observations. Maximum likelihood suffers the limited amount of observations. It shows a log-score that is really high compared to those of the other methods. Ridge, lasso and adaptive lasso do not suffer from the change in sample dimensions. Their log-scores follows the same trend observed when considering $n = 500$. Principal components regression shows an higher variability compared to ridge lasso and adaptive lasso. Principal components regression suffers a little bit the reduction of the sample dimensions.
With 500 observations we observe that as the correlation between the explanatory variables increases the variability in maximum likelihood log-scores tends to increase. Ridge reduces its variability as the correlation between the explanatory variables increases. Its log-scores tend to remain similar in all the scenarios. Lasso and adaptive lasso have quite the same log-scores re-

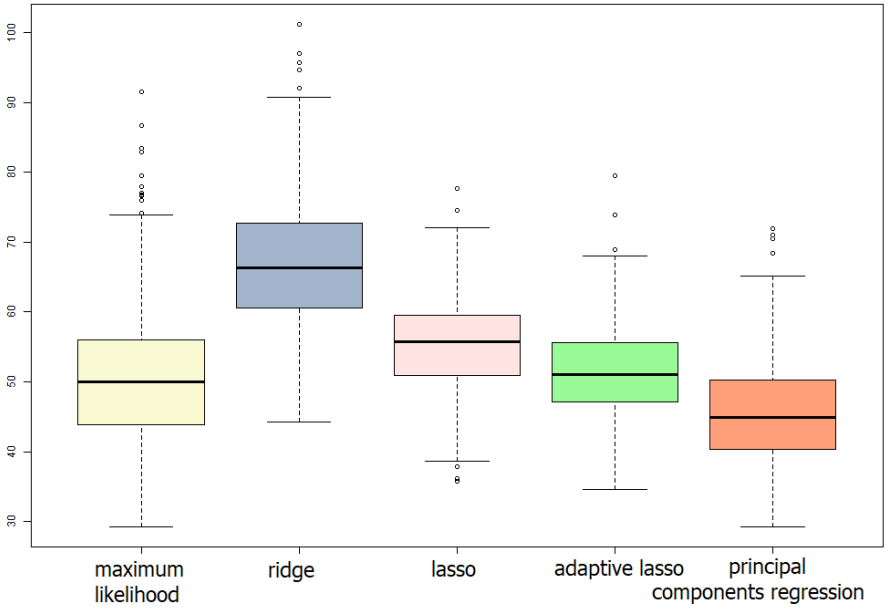

Figure 3.3: *Boxplots of the simulated log-scores for the fitting methods when the correlation parameter is equal to 0.6.*

gardless of the correlation parameter value. Finally, principal components regression tends to improve its log-score, also in terms of variability, as the correlation parameter value increases.
The four dimensionality reduction methods provide a better control on the variability of the data with respect to maximum likelihood, that is much more sensible to the variability, to the correlation between the explanatory variables and to the sample size.

3.2.2 Prediction Accuracy

Table 3.1 shows that as the correlation between the explanatory variables increases all methods improve their prediction accuracy. Maximum likelihood is the best method when the correlation parameter is equal to zero. Maximum likelihood as the correlation raises, despite the increased prediction accuracy, became the less accurate method. Ridge has a moderate prediction accuracy. Its prediction accuracy overcome the one of maximum likelihood only when the correlation parameter is equal to 0.9. Lasso always performs well and assumes high values of prediction accuracy. When the correlation parameter is equal to zero, the difference with respect to maximum likelihood is really

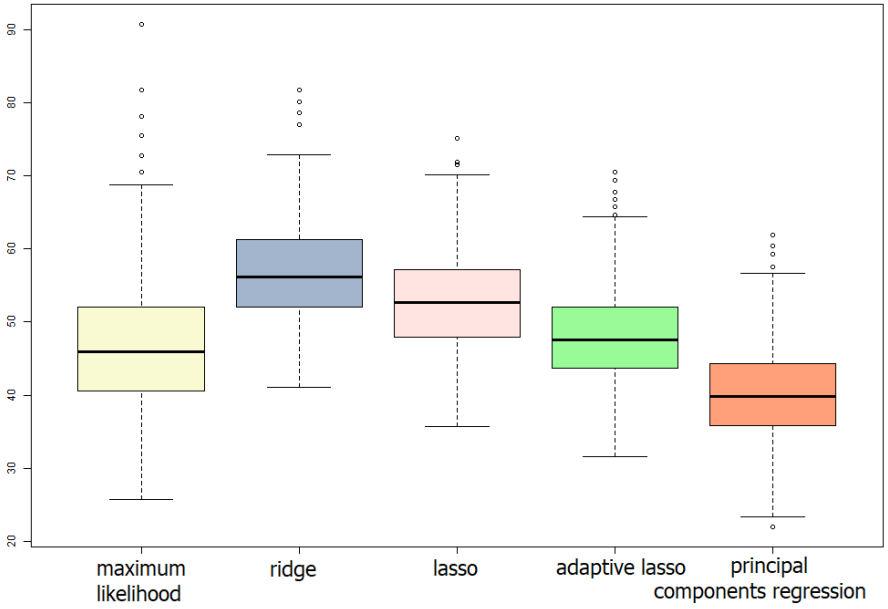

Figure 3.4: *Boxplots of the simulated log-scores for the fitting methods when the correlation parameter is equal to 0.9.*

Method	Corr 0	Corr 0.3	Corr 0.6	Corr 0.9
Maximum Likelihood	75.914	77.974	78.738	80.146
Ridge	74.656	76.869	78.68	80.621
Lasso	76.55	78.06	79.696	81.55
Adaptive Lasso	75.88	77.444	79.643	81.354
PCR	68.828	77.384	80.434	82.53

Table 3.1: *Prediction accuracy for all the performed scenarios.*

small. The prediction accuracy of adaptive lasso is lower compared to lasso but almost in all the scenarios adaptive lasso performs better than ridge and maximum likelihood. Finally, principal components regression shows a really poor prediction accuracy when the correlation parameter is equal to zero. As the correlation raises, the prediction accuracy of principal components regression improves and when the correlation parameter is equal or greater than 0.6 it becomes the best regression fitting method. The increasing of correlation between the explanatory variables improves the performances of

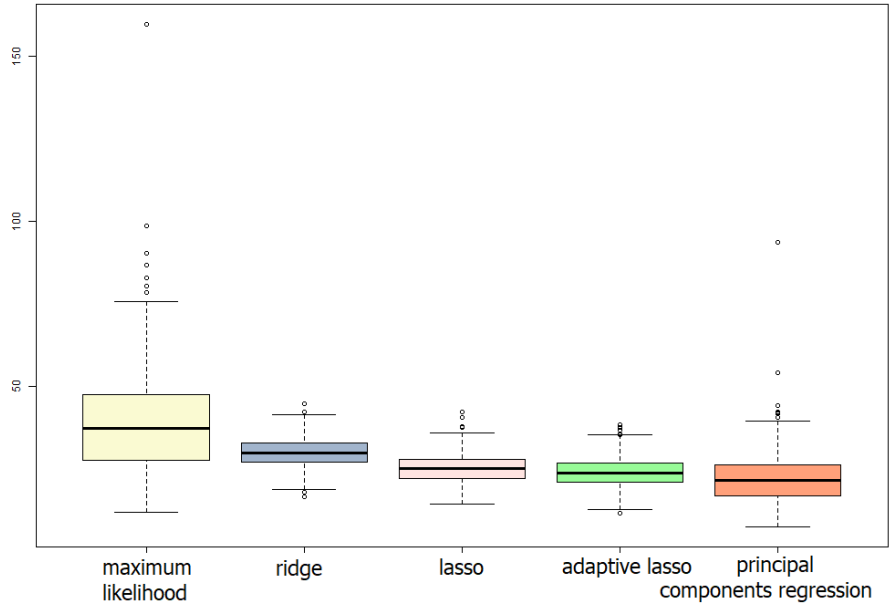

Figure 3.5: *Boxplots of the simulated log-scores for the fitting methods when the correlation parameter is equal to 0.6 and the sample size is n = 200.*

the dimensionality reduction methods, while in maximum likelihood causes a decreasing of prediction accuracy. The results in Figure 3.6 are equivalent to those reported in Table 3.1.

3.2.3 Estimated Coefficients

The results discussed below do not include principal components regression since its coefficients have a different meaning than those estimated by maximum likelihood, ridge regression and lasso.
The distribution of maximum likelihood estimates of β_1 coefficients is not centred around the true value, as shown in Figures 3.7, 3.10, 3.13 and 3.16. As the correlation between the explanatory variables increases, maximum likelihood variance tends to increase. Maximum likelihood estimates of β_9 are more centred around the true value, as shown if Figures 3.8, 3.11, 3.14 and 3.171. The variance raises as the correlation parameter value increases. Maximum likelihood estimates of β_{16} are centred on the true value, but there is an high variance.
Ridge estimates of β_1 tend to be more concentrated around zero as the cor-

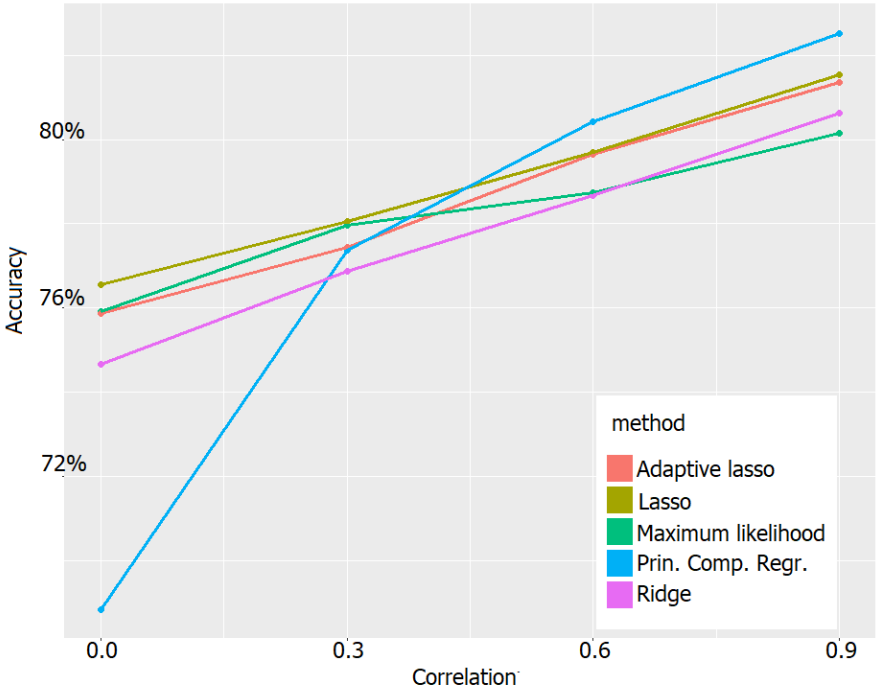

Figure 3.6: *Lines plot of the prediction accuracy for all regression fitting methods as the correlation parameter value increases.*

relation between the explanatory variables increases. Ridge estimates of β_9 tend to approach the true value as the correlation raises. Ridge estimates of β_{16} are always centred and concentrated towards the real value. The higher is the correlation parameter value, the more ridge estimates of β_{16} are concentrated around the true value. Lasso estimates of β_1, β_9 and β_{16} are similar to ridge. Lasso has an higher variance compared to ridge, as shown in Figures 3.11 or 3.16. Lasso estimates of β_{16} are more concentrated towards the real value with respect to ridge, since lasso shrinks the estimated coefficients exactly to zero. The distributions of adaptive lasso estimates is similar compared to ridge and lasso. However, we observe that adaptive lasso variance is higher in particular in correspondence of the estimates of β_1 and β_9, as shown in Figure 3.13. The adaptive lasso estimates of β_{16} are more similar to lasso distribution. However, in some cases we observe an higher number of outliers in adaptive lasso, as shown in Figure 3.12.

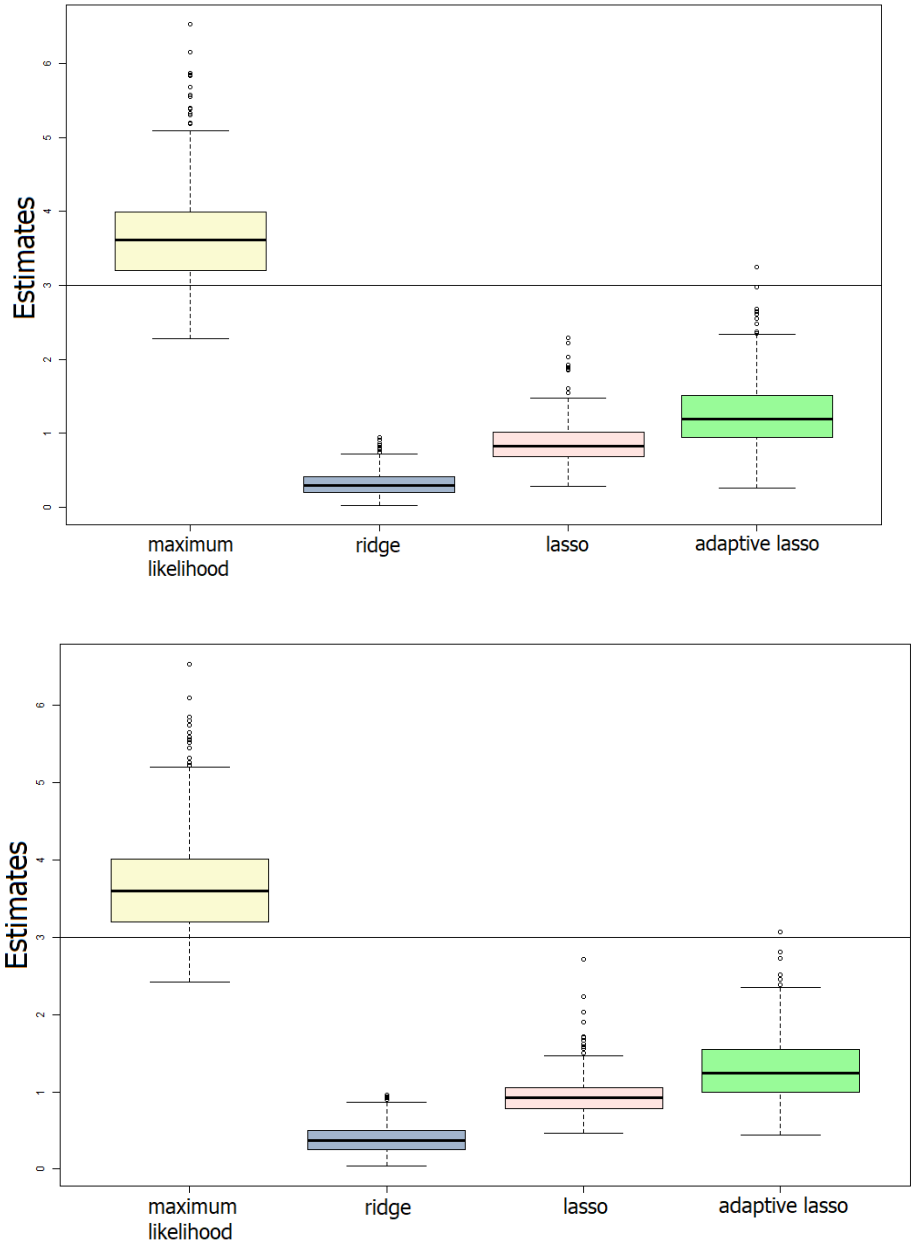

Figure 3.7: *Boxplots of the estimated coefficients for β_1 with the different fitting methods when the correlation parameter is equal to zero. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

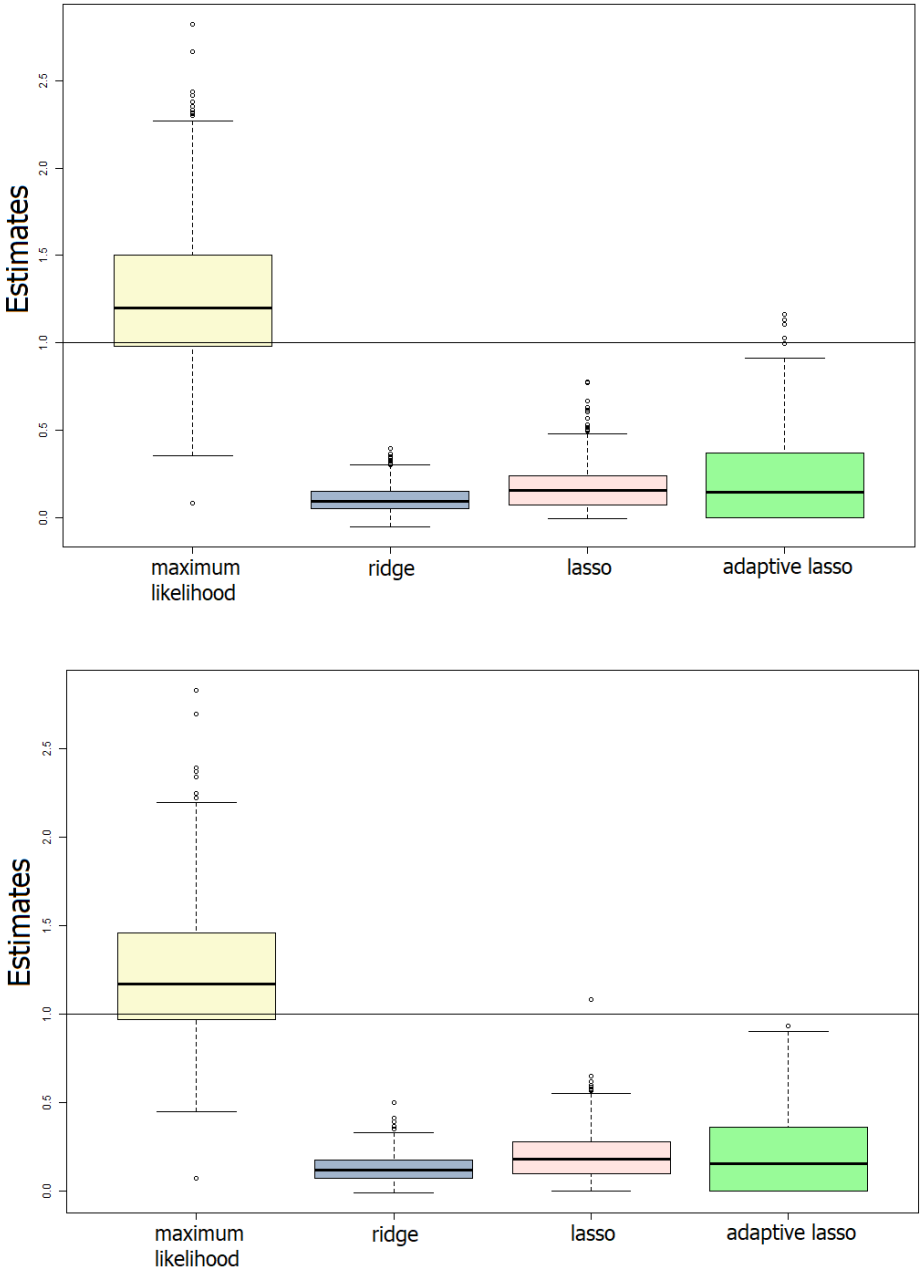

Figure 3.8: *Boxplots of the estimated coefficients for β_9 with the different fitting methods when the correlation parameter is equal to zero. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

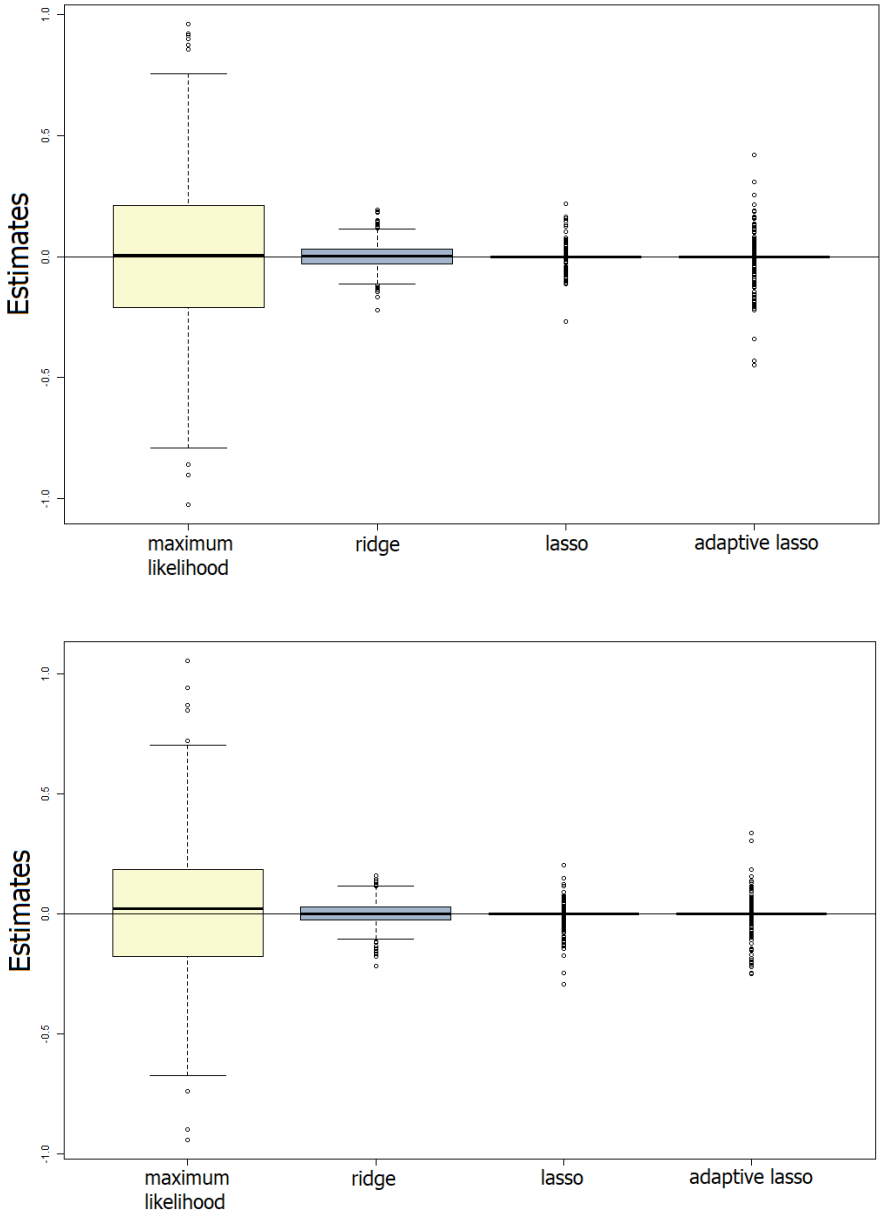

Figure 3.9: *Boxplots of the estimated coefficients for β_{16} with the different fitting methods when the correlation parameter is equal to zero. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

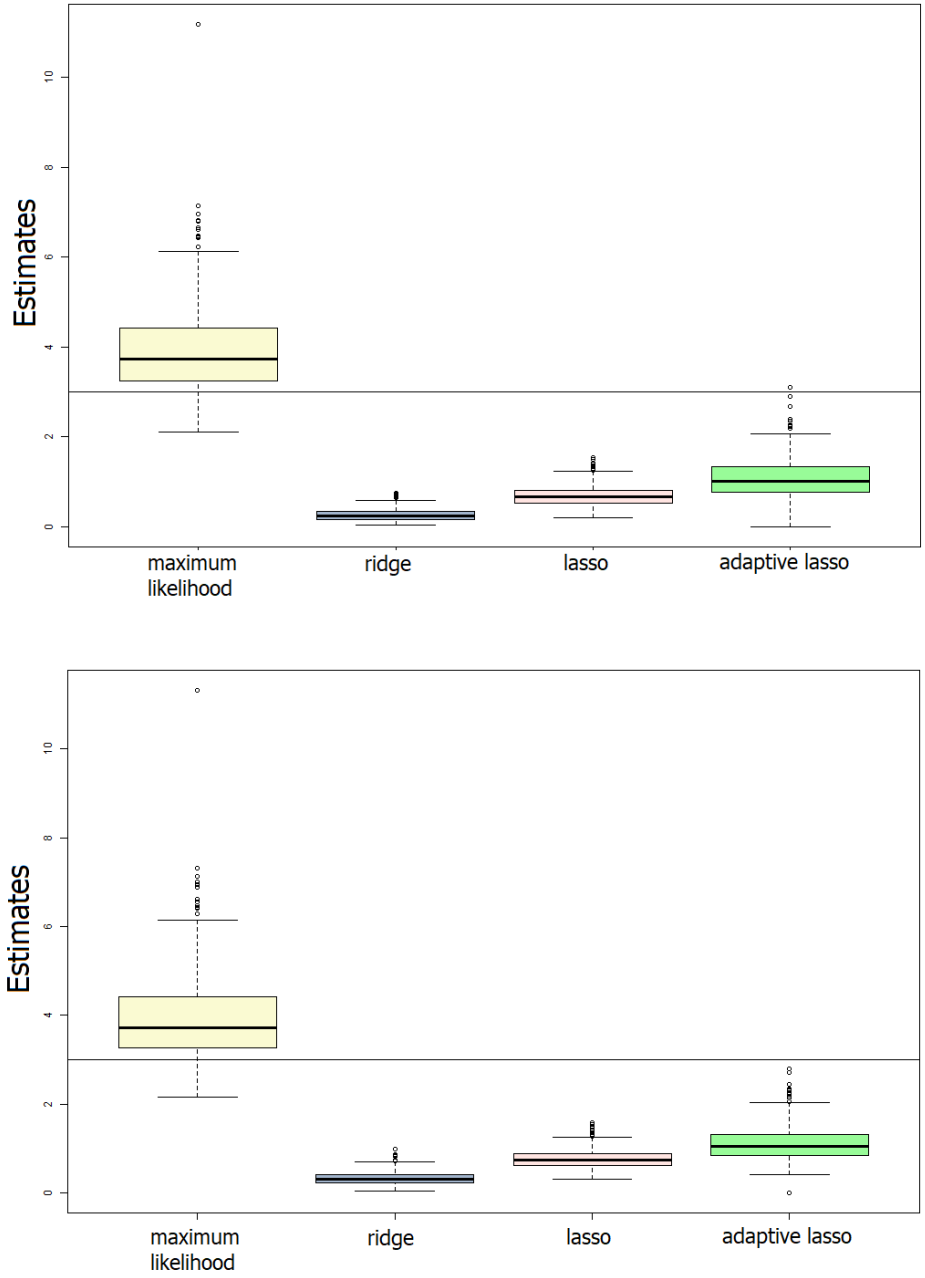

Figure 3.10: *Boxplots of the estimated coefficients for β_1 with the different fitting methods when the correlation parameter is equal to 0.3. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

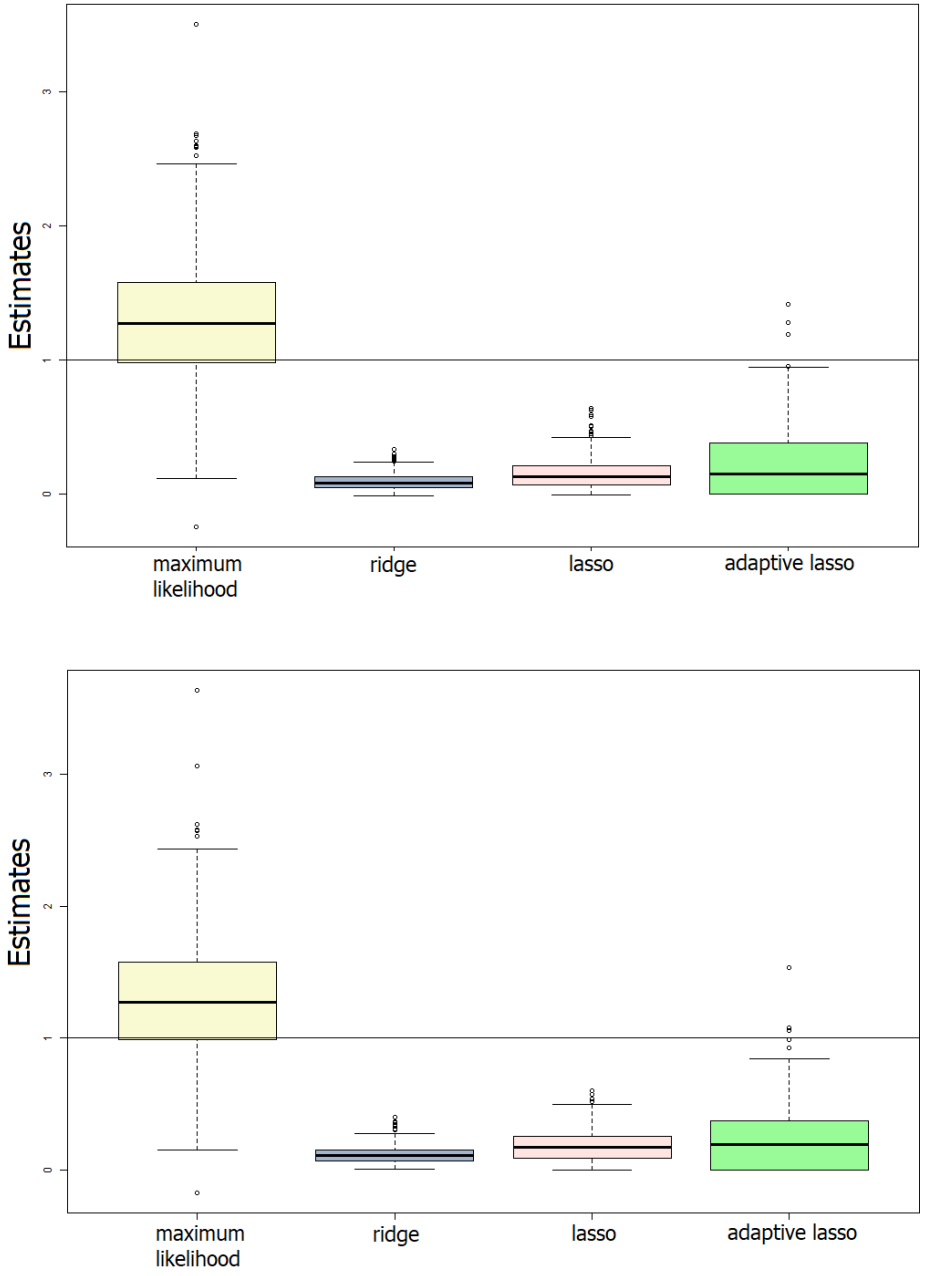

Figure 3.11: *Boxplots of the estimated coefficients for β_9 with the different fitting methods when the correlation parameter is equal to 0.3. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

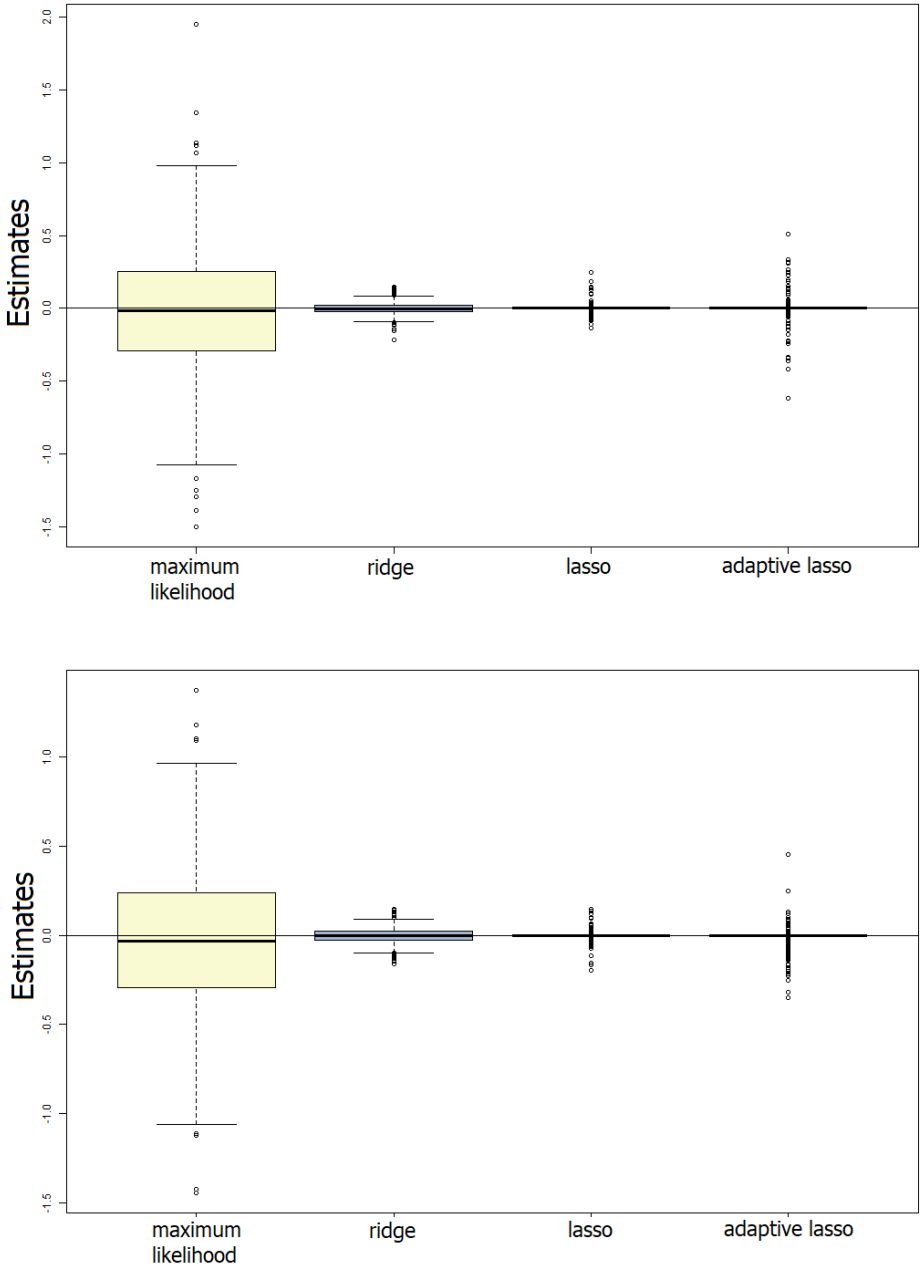

Figure 3.12: *Boxplots of the estimated coefficients for β_{16} with the different fitting methods when the correlation parameter is equal to 0.3. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

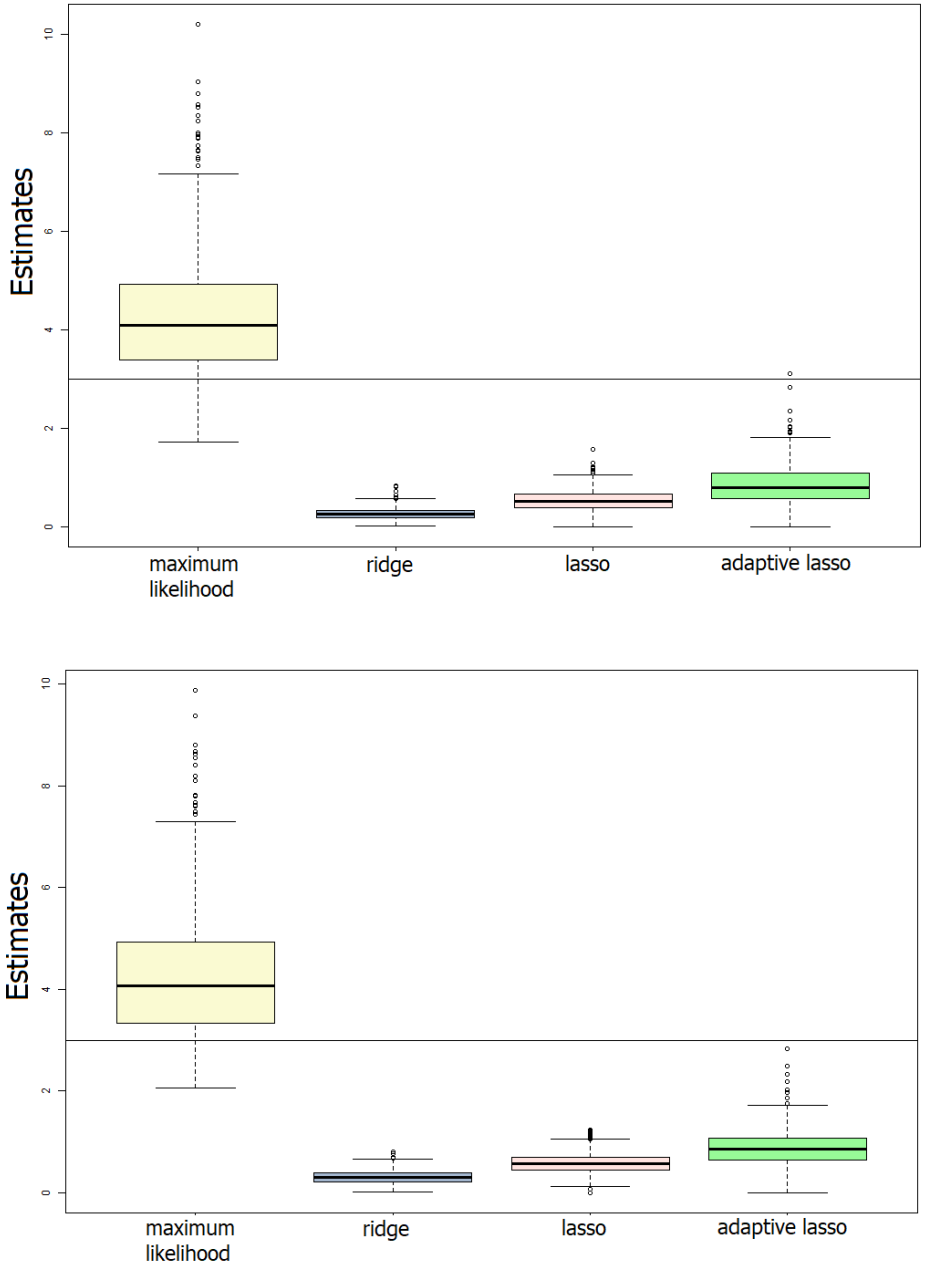

Figure 3.13: *Boxplots of the estimated coefficients for β_1 with the different fitting methods when the correlation parameter is equal to 0.6. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

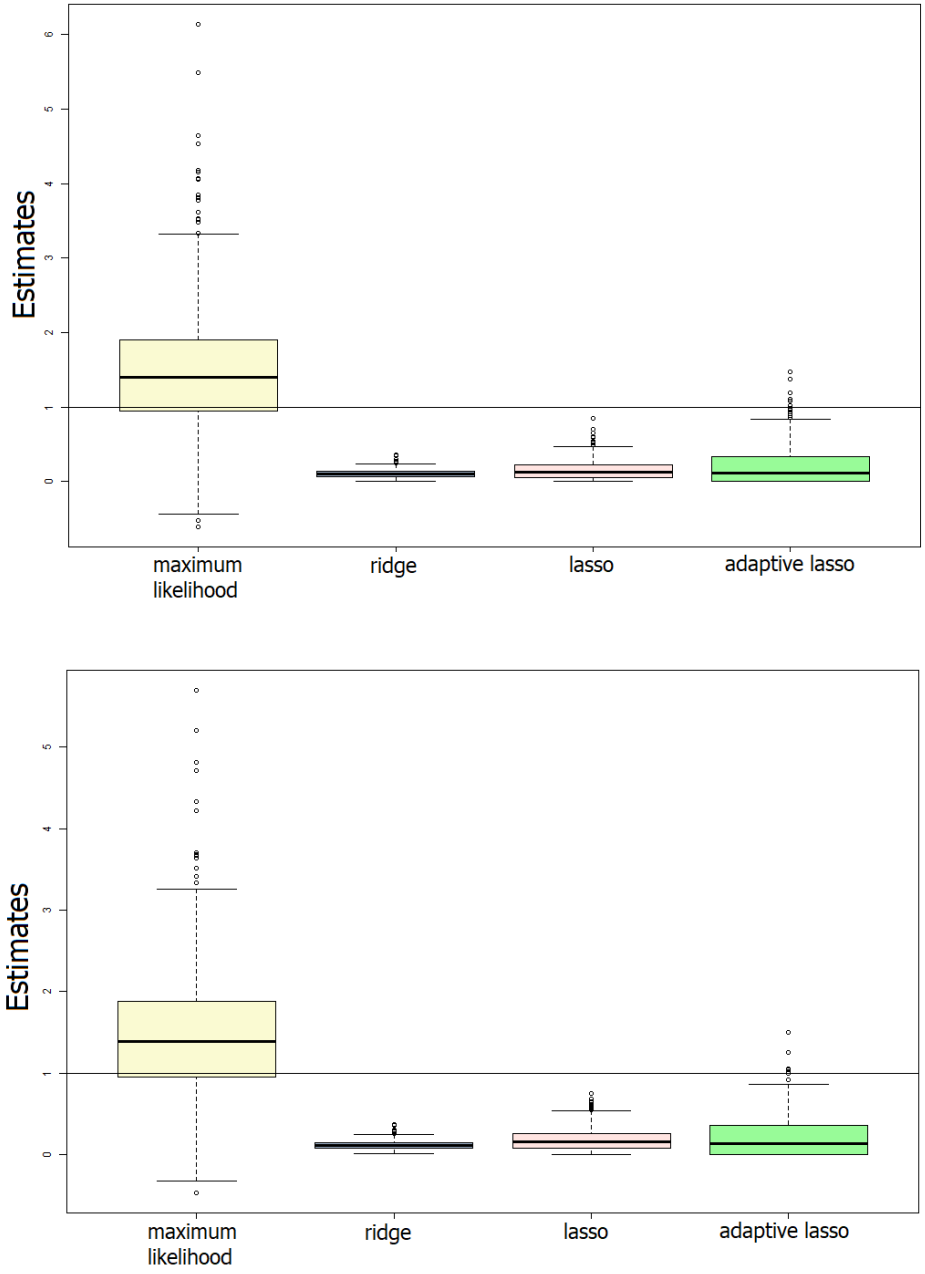

Figure 3.14: *Boxplots of the estimated coefficients for β_9 with the different fitting methods when the correlation parameter is equal to 0.6. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

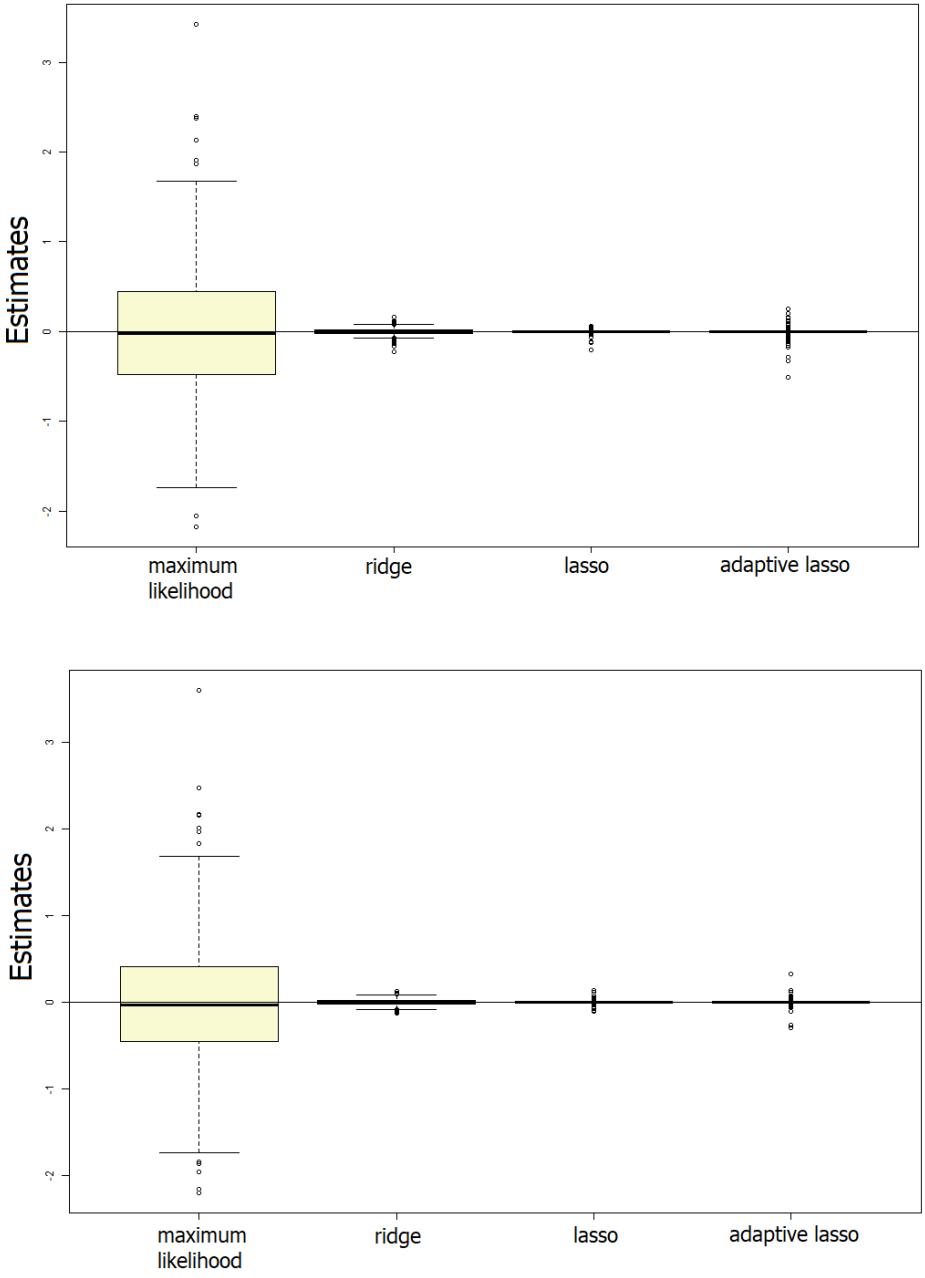

Figure 3.15: *Boxplots of the estimated coefficients for* β_{16} *with the different fitting methods when the correlation parameter is equal to 0.6. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

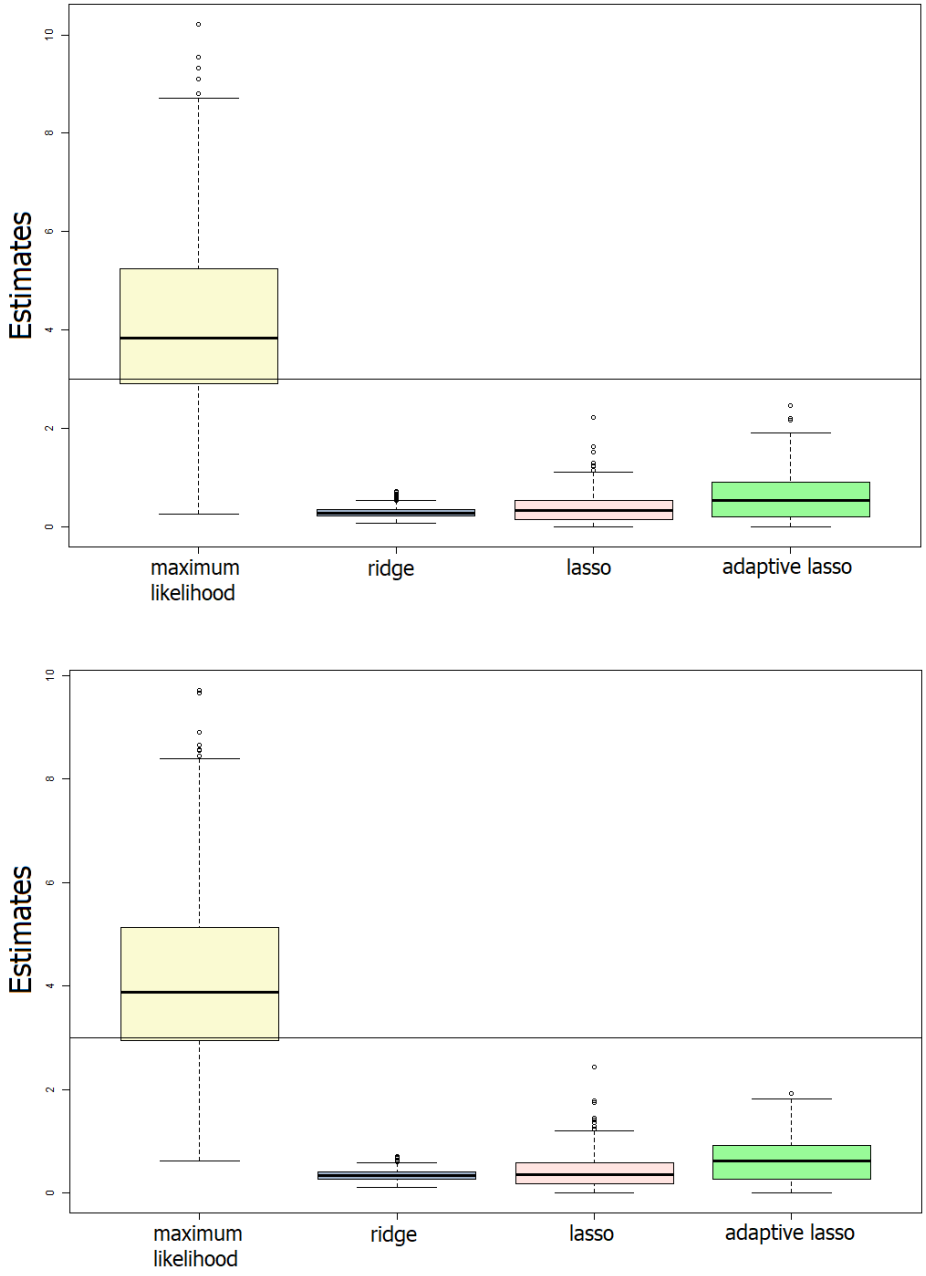

Figure 3.16: *Boxplots of the estimated coefficients for β_1 with the different fitting methods when the correlation parameter is equal to 0.9. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

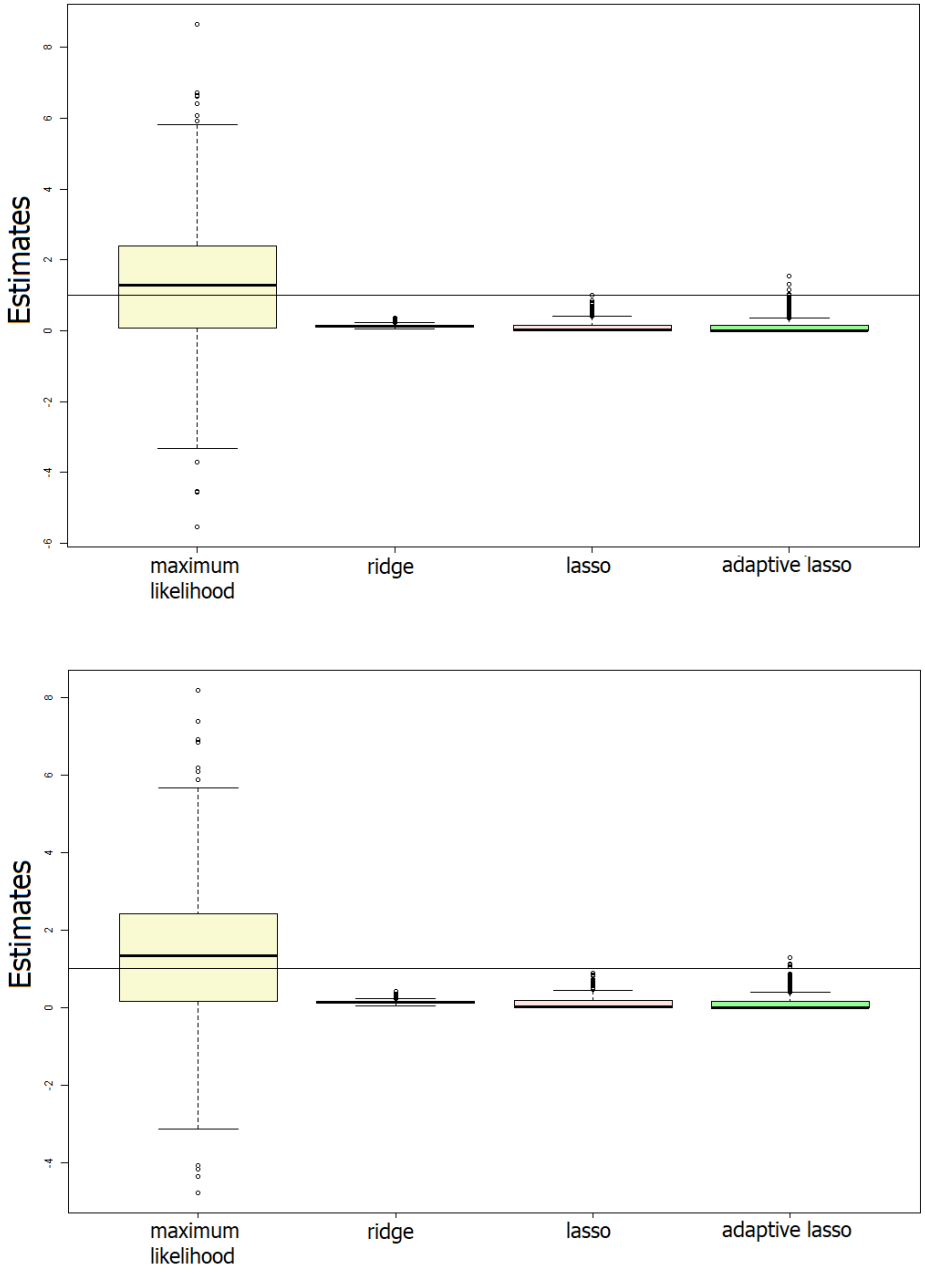

Figure 3.17: *Boxplots of the estimated coefficients for β_9 with the different fitting methods when the correlation parameter is equal to 0.9. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

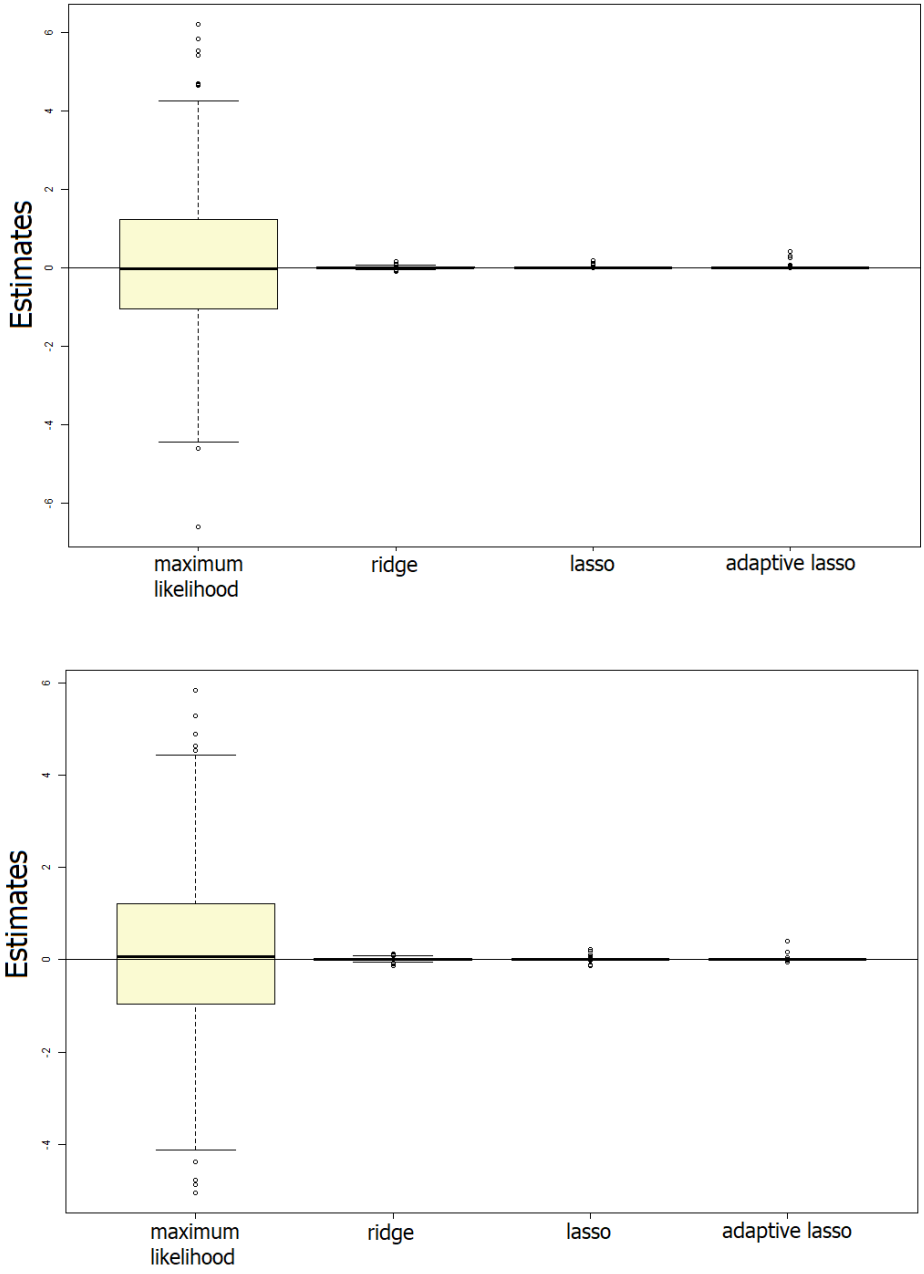

Figure 3.18: *Boxplots of the estimated coefficients for β_{16} with the different fitting methods when the correlation parameter is equal to 0.9. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

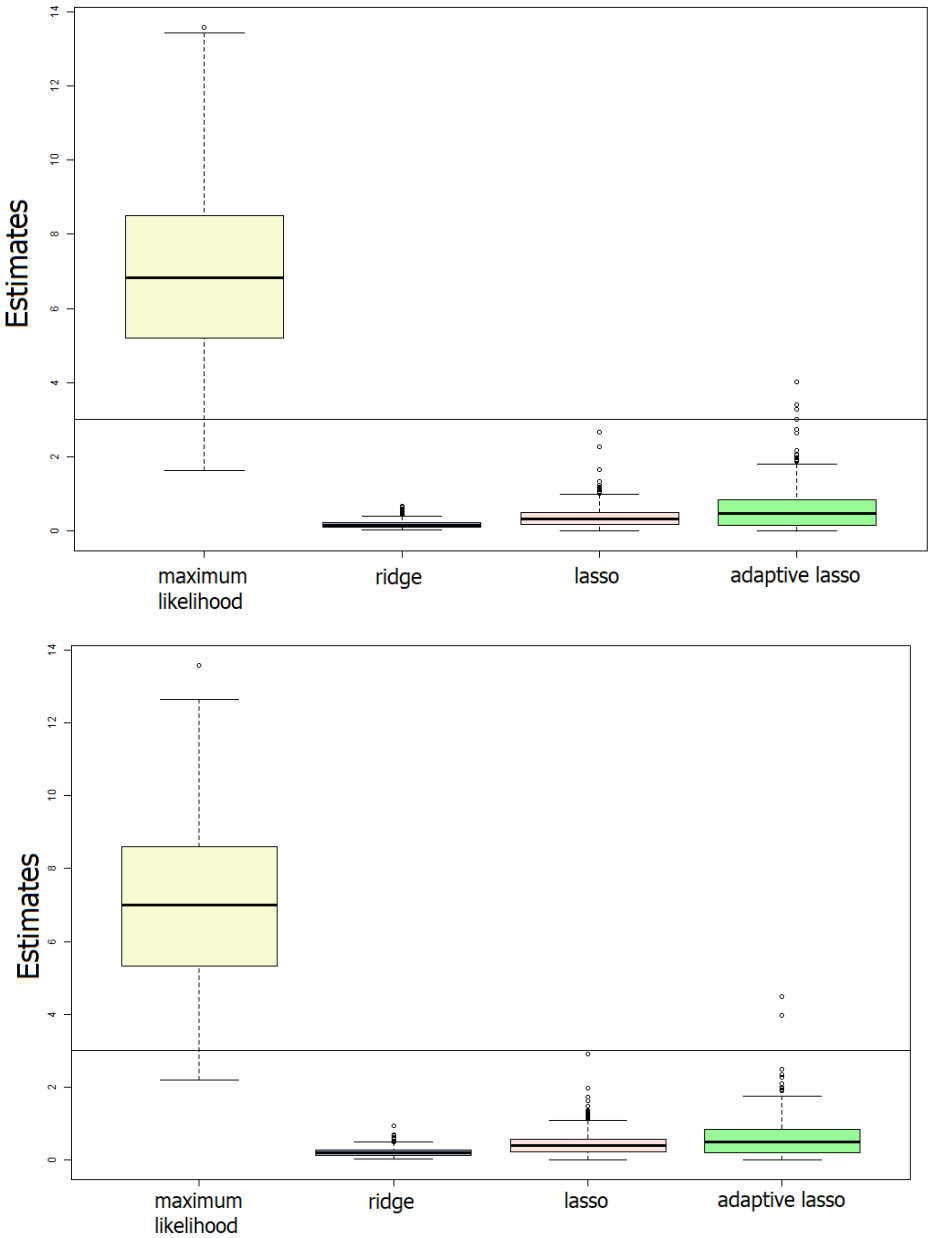

Figure 3.19: *Boxplots of the estimated coefficients for β_1 with the different fitting methods when the correlation parameter is equal to 0.6 and the sample size is equal to $n = 200$. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

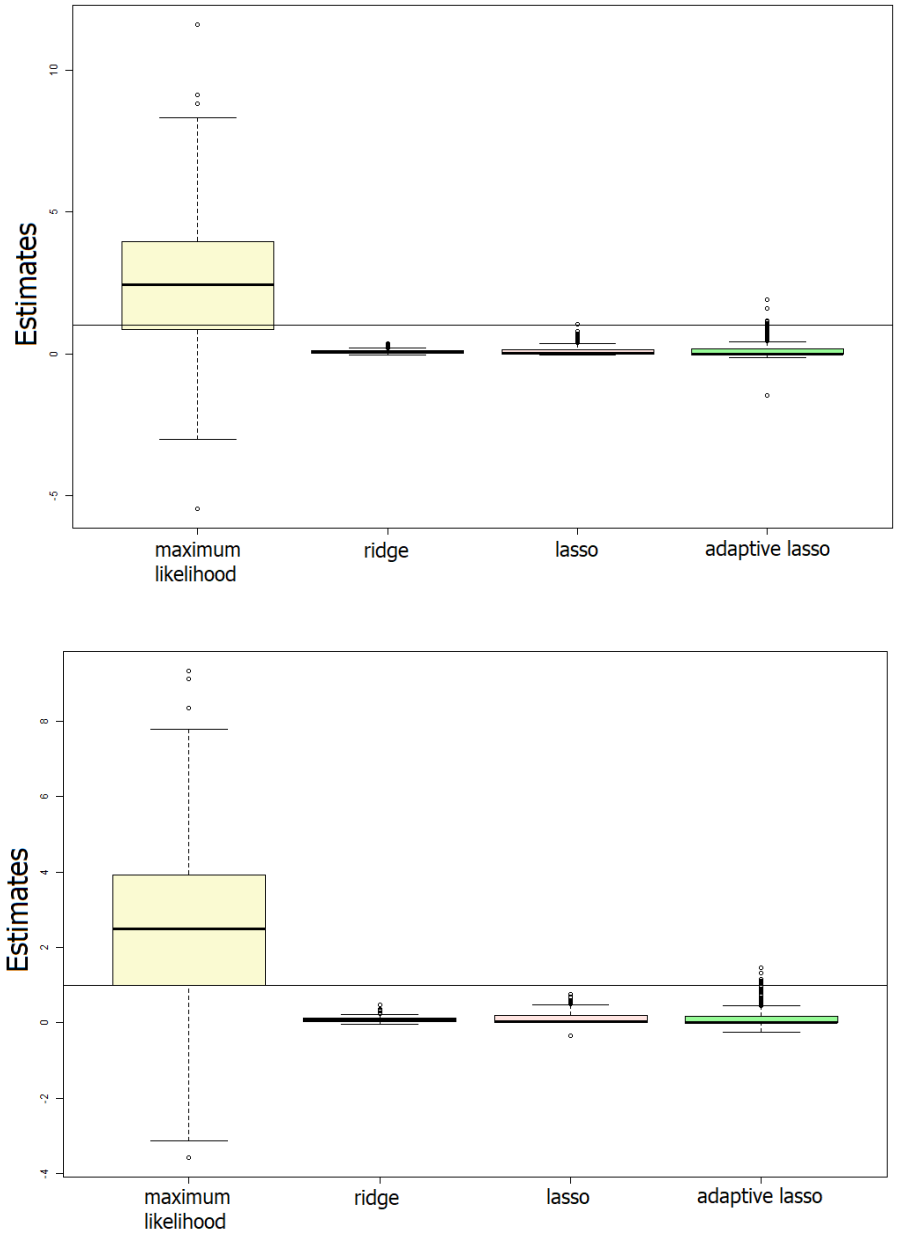

Figure 3.20: *Boxplots of the estimated coefficients for β_9 with the different fitting methods when the correlation parameter is equal to 0.6 and the sample size is equal to $n = 200$. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

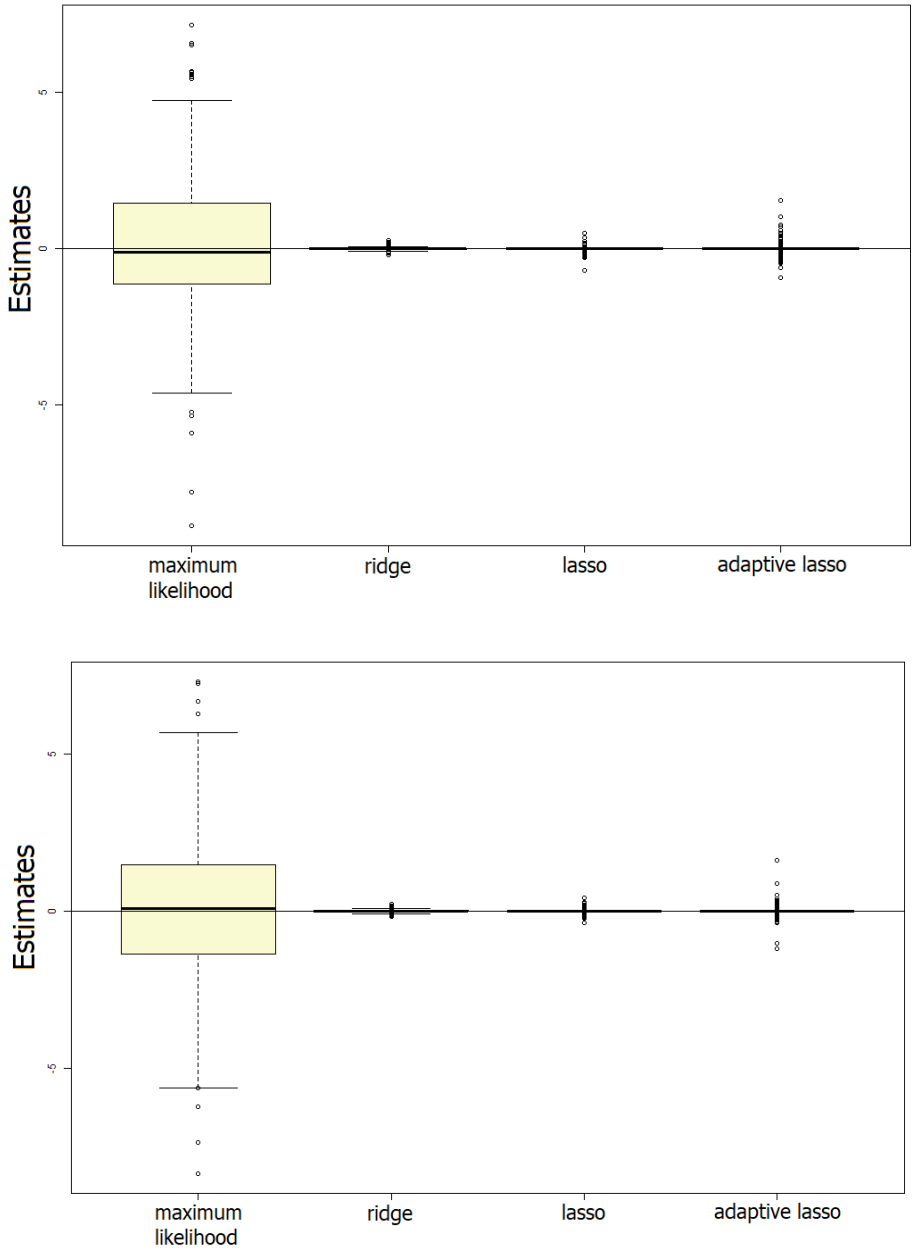

Figure 3.21: *Boxplots of the estimated coefficients for β_{16} with the different fitting methods when the correlation parameter is equal to 0.6 and the sample size is equal to $n = 200$. Left panel: coefficients for category 1. Right panel: coefficients for category 2. The horizontal line correspond to the true value.*

3.2.4 Lasso Selection

The explanatory variables that we expect the two lasso methods will select are those for which $\beta = 3$ and $\beta = 1$. The explanatory variables that should be shrunk to zero are those for which $\beta = 0$. When the correlation parameter is equal to zero as shown in Table 3.2, lasso has a very good performance in selecting the truly non-null explanatory variables and exclude that are really irrelevant. Adaptive lasso has a lower accuracy with respect to lasso either in selecting the important explanatory variables and excluding the non-important ones. This is observable in all the scenarios shown in Tables 3.3, 3.4, and 3.5. As the correlation increases the performance in selecting the explanatory variables decreases in both methods. When the correlation parameter value raises, lasso and adaptive lasso tend to exclude more the explanatory variables rather than selecting them.

Method	**Correct selection (%)**	**Correct exclusion (%)**
Lasso	95.78	75.04
Adaptive lasso	83.24	75.50

Table 3.2: *Lasso and adaptive lasso accuracy in selecting the explanatory variables when the correlation parameter is equal to zero.*

Method	**Correct selection (%)**	**Correct exclusion (%)**
Lasso	96.04	84.78
Adaptive lasso	83.38	81.16

Table 3.3: *Lasso and adaptive lasso accuracy in selecting the explanatory variables when the correlation parameter is equal to 0.3.*

Method	**Correct selection (%)**	**Correct exclusion (%)**
Lasso	92.20	90.06
Adaptive lasso	80.76	90.08

Table 3.4: *Lasso and adaptive lasso accuracy in selecting the explanatory variables when the correlation parameter is equal to 0.6.*

3.2.5 Root Mean Square Errors

Since the estimated coefficients from the principal components regression have a different meaning from those of the other methods we exclude the

Method	Correct selection (%)	Correct exclusion (%)
Lasso	76.72	94.68
Adaptive lasso	65.84	96.78

Table 3.5: *Lasso and adaptive lasso accuracy in selecting the explanatory variables when the correlation parameter is equal to 0.6.*

principal components regression from the comparisons in terms of root mean square error discussed below.
Maximum likelihood root mean square error is lower compared to the other methods if we consider the first ten estimated coefficients that are truly non-zero. The other three methods that perform shrinkage on the estimated coefficients show an higher bias looking at the first ten $\hat{\beta}$ coefficients. On the other hand as shown in Tables 3.6, 3.7, 3.8, and 3.9, the root mean square error for the last ten estimated coefficients is lower in ridge, lasso and adaptive lasso, compared to maximum likelihood $\hat{\beta}$ coefficients.
In general, as the correlation raises, we observe an increasing of root mean square error for all the methods, for the first ten estimated coefficients. Ridge, lasso and adaptive lasso root mean square errors decreases in the last ten $\hat{\beta}$ coefficients as the correlation rises.

3.2.6 Selected Principal Components

We considered two scenarios to describe principal components regression. The first scenario takes into account a sample with $n = 200$ observations, the second a sample with $n = 500$ observations. In both scenarios the correlation parameter value is equal to 0.6. The number of principal components selected with $n = 200$ oscillates between ten and eleven. When the number of observations is equal to $n = 500$ the number of selected principal components stabilize to eleven.
We expect that principal components regression selects a number of principal components inferior to the total number of explanatory variables in the design matrix. As we can see in Table 3.10 when the correlation parameter value is equal to zero, the selected principal components are 14, that is a discrete reduction in dimensionality compared to the original 20 explanatory variables. As the correlation rises the number of selected principal components decreases. Table 3.10 shows that when the correlation parameter is equal to 0.9 the principal components regression selects seven principal components, a remarkable reduction of dimensionality.

	Maximum likelihood		Ridge		Lasso		Adaptive lasso	
# variable	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2
1	0.946	0.941	2.687	2.619	2.153	2.073	1.816	1.766
2	0.99	0.987	2.682	2.615	2.146	2.068	1.813	1.762
3	0.96	0.951	2.682	2.618	2.145	2.07	1.802	1.761
4	0.982	0.986	2.687	2.619	2.147	2.068	1.825	1.768
5	0.986	0.986	2.683	2.617	2.148	2.069	1.818	1.769
6	0.446	0.427	0.895	0.875	0.847	0.82	0.82	0.817
7	0.45	0.434	0.894	0.873	0.844	0.816	0.815	0.812
8	0.442	0.415	0.895	0.876	0.844	0.822	0.823	0.821
9	0.471	0.451	0.895	0.873	0.84	0.815	0.82	0.819
10	0.456	0.414	0.895	0.877	0.846	0.823	0.823	0.821
11	0.332	0.31	0.055	0.05	0.037	0.035	0.07	0.046
12	0.339	0.299	0.054	0.051	0.042	0.034	0.067	0.047
13	0.327	0.285	0.054	0.047	0.028	0.027	0.07	0.033
14	0.343	0.319	0.051	0.053	0.03	0.034	0.071	0.058
15	0.324	0.296	0.055	0.049	0.031	0.031	0.064	0.046
16	0.321	0.295	0.053	0.05	0.03	0.034	0.064	0.049
17	0.322	0.303	0.053	0.054	0.032	0.039	0.068	0.049
18	0.336	0.301	0.056	0.051	0.04	0.036	0.07	0.048
19	0.316	0.292	0.053	0.05	0.026	0.03	0.057	0.042
20	0.34	0.305	0.056	0.053	0.035	0.036	0.083	0.056

Table 3.6: *Root mean square error for the fitting methods when the correlation parameter is equal to zero.*

	Maximum likelihood		Ridge		Lasso		Adaptive lasso	
# variable	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2
1	1.337	1.346	2.739	2.682	2.314	2.242	1.981	1.935
2	1.341	1.344	2.706	2.648	2.249	2.184	1.891	1.849
3	1.371	1.368	2.697	2.644	2.245	2.188	1.881	1.856
4	1.36	1.352	2.703	2.649	2.243	2.185	1.867	1.841
5	1.351	1.354	2.73	2.675	2.263	2.193	1.907	1.852
6	0.576	0.555	0.871	0.85	0.834	0.812	0.793	0.789
7	0.611	0.591	0.889	0.875	0.838	0.818	0.805	0.802
8	0.598	0.59	0.897	0.877	0.842	0.82	0.786	0.784
9	0.581	0.56	0.906	0.885	0.855	0.827	0.814	0.811
10	0.606	0.586	0.915	0.899	0.887	0.865	0.855	0.855
11	0.445	0.406	0.055	0.047	0.026	0.024	0.074	0.053
12	0.469	0.419	0.048	0.042	0.023	0.025	0.056	0.045
13	0.416	0.392	0.045	0.044	0.024	0.031	0.061	0.038
14	0.432	0.402	0.047	0.04	0.022	0.02	0.058	0.042
15	0.45	0.426	0.043	0.042	0.02	0.019	0.047	0.032
16	0.437	0.419	0.044	0.043	0.023	0.023	0.069	0.048
17	0.429	0.405	0.044	0.041	0.029	0.026	0.059	0.051
18	0.427	0.401	0.045	0.043	0.025	0.026	0.053	0.043
19	0.455	0.418	0.045	0.046	0.022	0.023	0.059	0.039
20	0.442	0.412	0.044	0.043	0.031	0.024	0.063	0.037

Table 3.7: *Root mean square error for the fitting methods when the correlation parameter is equal to 0.3.*

	Maximum likelihood		Ridge		Lasso		Adaptive lasso	
# variable	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2
1	1.801	1.794	2.735	2.69	2.461	2.413	2.185	2.154
2	1.785	1.775	2.694	2.647	2.34	2.295	2.022	1.995
3	1.838	1.85	2.681	2.631	2.324	2.272	2.009	1.978
4	1.861	1.851	2.69	2.641	2.331	2.288	2.036	2.005
5	1.861	1.847	2.724	2.681	2.337	2.298	2.002	1.97
6	0.95	0.928	0.825	0.807	0.832	0.822	0.767	0.766
7	0.956	0.915	0.863	0.85	0.841	0.829	0.799	0.797
8	0.925	0.903	0.881	0.867	0.837	0.818	0.812	0.807
9	0.998	0.968	0.893	0.88	0.851	0.833	0.836	0.829
10	0.909	0.888	0.92	0.905	0.928	0.915	0.909	0.91
11	0.775	0.73	0.055	0.056	0.023	0.031	0.088	0.059
12	0.765	0.732	0.043	0.045	0.031	0.03	0.05	0.035
13	0.759	0.738	0.039	0.039	0.017	0.025	0.04	0.032
14	0.759	0.741	0.038	0.039	0.008	0.012	0.039	0.025
15	0.713	0.701	0.041	0.04	0.012	0.019	0.037	0.024
16	0.744	0.739	0.039	0.037	0.015	0.016	0.04	0.027
17	0.743	0.728	0.036	0.036	0.013	0.011	0.054	0.027
18	0.722	0.693	0.038	0.038	0.022	0.016	0.038	0.027
19	0.731	0.718	0.04	0.038	0.016	0.019	0.043	0.031
20	0.639	0.626	0.043	0.042	0.027	0.023	0.041	0.034

Table 3.8: *Root mean square error for the fitting methods when the correlation parameter is equal to 0.6.*

	Maximum likelihood		Ridge		Lasso		Adaptive lasso	
# variable	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2	Categ 1	Categ 2
1	2.058	2.01	2.699	2.66	2.645	2.617	2.442	2.424
2	2.194	2.152	2.686	2.644	2.491	2.45	2.255	2.214
3	2.214	2.148	2.683	2.643	2.464	2.428	2.223	2.185
4	2.232	2.197	2.694	2.658	2.461	2.435	2.207	2.175
5	2.156	2.129	2.721	2.693	2.489	2.466	2.253	2.225
6	1.959	1.895	0.771	0.757	0.84	0.832	0.786	0.774
7	1.817	1.789	0.808	0.794	0.855	0.841	0.833	0.821
8	1.789	1.731	0.838	0.827	0.876	0.863	0.879	0.874
9	1.901	1.872	0.862	0.853	0.905	0.898	0.908	0.905
10	1.917	1.905	0.888	0.884	0.951	0.946	0.956	0.957
11	1.798	1.752	0.085	0.087	0.054	0.068	0.094	0.082
12	1.802	1.709	0.064	0.063	0.027	0.027	0.053	0.034
13	1.798	1.742	0.049	0.048	0.023	0.024	0.013	0.009
14	1.863	1.794	0.041	0.04	0.016	0.011	0.03	0.016
15	1.901	1.836	0.036	0.035	0.017	0.013	0.022	0.013
16	1.84	1.739	0.033	0.033	0.012	0.016	0.026	0.02
17	1.817	1.788	0.034	0.032	0.018	0.02	0.014	0.014
18	1.916	1.883	0.032	0.03	0.021	0.039	0.03	0.018
19	1.805	1.804	0.034	0.033	0.015	0.017	0.024	0.02
20	1.274	1.269	0.037	0.038	0.023	0.022	0.021	0.013

Table 3.9: *Root mean square error for the fitting methods when the correlation parameter is equal to 0.9.*

Correlation	Principal components
0.0	14
0.3	13
0.6	11
0.9	7

Table 3.10: *Number of selected principal components when the sample size is equal to $n = 500$, as a function of the correlation parameter ρ.*

Chapter 4

New York Police Department Crimes Data

The case study discussed in this chapter is based on an open dataset retrieved from the NYC OpenData website (NYCOpenData, 2018). This archive collects a huge amount of open data that the New York City authorities provide to the public. The dataset includes the crimes that the New York Police Department reported from 1972 to 2018. The objective is to build a multinomial logistic regression model for prediction of the "crime category" classified as felony, misdemeanor or violation. The prediction is made on the basis of the information contained in a series of explanatory variables that are the characteristics of the victims and the suspects, the place in which the complaint took place, the description of the crime and the competent jurisdiction.

4.1 Case Study Definition

For the majority of the reported years the amount of missing data is substantial so that we do not have enough information to perform a good data analysis. Thus we took into account only the year 2018 for which have been provided a satisfactory crimes description. We aimed to predict the crime categories that the New York Police Department assign to the different crimes. The crime category is classified in three levels:

1. Felony crimes;
2. Misdemeanor crimes;
3. Violation crimes.

In addition to the crime category, the dataset includes various explanatory variables:

- *Complaint from date*: The date of occurrence of the crime;
- *Complaint time from*: The time of occurrence of the crime;
- *Complaint to date*: The ending date of occurrence of the crime;
- *Complaint to time*: The ending time of occurrence of the crime;
- *Reported date*: The crime date reported to the police;
- *Key Code*: Three digits code to classify the crime;
- *Key code offence description*: Description of the crime, corresponds to the crime code;
- *Police department code*: A three digits code used by the police department to internally classify the crime;
- Police department description: The description of the crime that corresponds to the internal classification code provided by the police department;
- *Crime Complete*: The indication if the crime have been completed, attempted or prematurely interrupted;
- *Jurisdiction description*: The jurisdiction responsible for the reported crime;
- *Borough name*: The name of the borough in which the crime occurs;
- *Precinct code*: The precinct code in which the crime occurs;
- *Location of occurrence description*: The specific location of occurrence of the crime, in or around the premises;
- *Premises type description*: The description of the premises at which the crime occurs;
- *Parks name*: If applicable, the name of the park at which the crime occurs;
- *Authority development*: The name of the housing authority development of occurrence, if applicable to the crime;

- *Suspect age group*: The group of age of the suspect (for example 18-24);
- *Suspect race*: The ethnicity of the crime suspect;
- *Suspect sex*: The sex of the crime suspect;
- *Victim age group*: The group of age of the crime victim (for example 18-24);
- *Victim race*: The ethnicity of the crime victim;
- *Victim sex*: The sex of the crime victim;
- *X-coordinate*: X-coordinate for the New York State Plane Coordinate System;
- *Y-coordinate*: Y-coordinate for the New York State Plane Coordinate System;
- *Latitude*:The latitude coordinate for the Global Coordinate System, decimal degrees;
- *Longitude*: The longitude coordinate for the Global Coordinate System, decimal degrees.

From these explanatory variables we selected a subset used to perform the multinomial logistic regression. We decided to use only the subset of explanatory variables that are of higher interest for the problem. The selected explanatory variables are:

- Location of occurrence;
- Borough name;
- Crime Complete;
- Jurisdiction description;
- Suspect race;
- Suspect sex;
- Victim race;
- Victim sex.

The experiment takes into account as train set the months of Jenuary,February, March and April 2018. We then made predictions on the test set that contains the crimes reported in May and June 2018. The train set is defined by 5677 observations and the test set by 2839 observations. Since some of the levels of the explanatory variables have substantially small number of observations, we dropped them from the analysis. The sample size is equal to 8516 observations and 20 explanatory variables.
The frequency distribution of the response variable *law category* is described in Table 4.1 and Figure 4.1. We observe that the most observed crime category is the *misdemeanor* with 3884 observations. The other two categories have quite similar frequencies, the *felony* crimes are 2229 and the *violations* crimes are 2403.

Law category	**Frequency**
Felony	2229
Misdemeanor	3884
Violation	2403

Table 4.1: *Distribution of the response variable law category.*

The distribution of *Location of occurrence* is described in Table 4.2. The majority of the crimes occur inside the location. We observe that the first two levels have high frequency, while *Opposite of* and *Rear of* frequencies are substantially smaller.

Location of occurrence	**Frequency**
Front of	1917
Inside	6406
Opposite of	105
Rear of	88

Table 4.2: *Distribution of the response variable law category.*

The distribution of *Borough name* is described in Table 4.3. We see that the borough in which the crimes frequency is higher is Brooklyn, while the lower frequency corresponds to *Staten Island*.

The distribution of *Crime complete* is summarized in Table 4.4, and we observe that almost all the reported crimes have been completed.

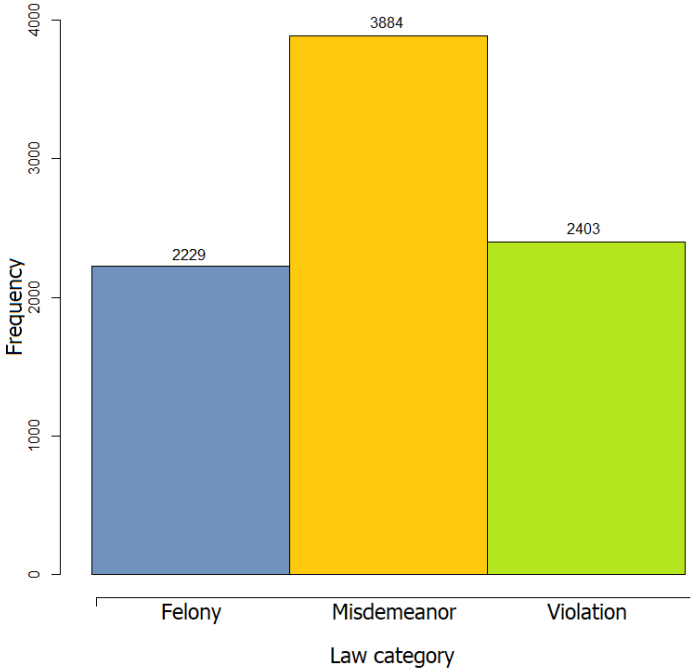

Figure 4.1: *The response variable frequencies.*

Borough	**Frequency**
Bronx	2208
Brooklyn	2530
Manhattan	1798
Queens	1544
Staten Island	436

Table 4.3: *Distribution of the explanatory variable Borough.*

Crime Complete	**Frequency**
Completed	8387
Not completed	129

Table 4.4: *Distribution of the explanatory variable Crime complete.*

The distribution of *Jurisdiction description* (Table 4.5) shows that the most observed level is the *N.Y. police department.*

The distribution of *Suspect race* described in Table 4.6 shows the majority

Jurisdiction description	Frequency
N.Y. Housing Police	986
N.Y. Police Dept.	7530

Table 4.5: *Distribution of the explanatory variable Crime complete.*

of observation if the suspect ethnicity is *black* or *unknown* or *white Hispanic*.

Suspect race	Frequency
Black	3255
Black Hispanic	534
Unknown	2144
White	906
White Hispanic	1677

Table 4.6: *Distribution of the explanatory variable Suspect race.*

Table 4.7 describe the distribution of *Suspect sex*. They show that the majority of the crimes are committed by men, more than the double with respect the crimes committed by women. The third level corresponds to an unknown gender of the suspect.

Suspect sex	Frequency
F	1935
M	4909
U	1672

Table 4.7: *Distribution of the explanatory variable Suspect sex.*

The distribution of *Victim race* is described in Table 4.8. It shows the majority of observations in correspondence of the level *black*. From what has been observed in the case of the suspect ethnicity we can say that there is an high number of black people that commit crimes, but also a large portion of the population that is victim of crimes is often a black person.

Table 4.9 shows the two levels of the explanatory variable *Victim sex*. We can observe that in the most of the cases the victims of the committed crimes are the women with 5224 observations.

Victim race	Frequency
Black	3389
Black Hispanic	525
Unknown	526
White	1753
White Hispanic	2323

Table 4.8: *Distribution of the explanatory variable Victim race.*

Suspect sex	Frequency
F	5224
M	3292

Table 4.9: *Distribution of the explanatory variable Victim sex.*

It is useful to summarize the joint distribution of the response with each of the considered explanatory variables. Table 4.10 we observe that the majority of the misdemeanor crimes are committed inside the location. In general, there is an higher frequency of felony and violation crimes committed inside the location. There are cases in which the crime is committed in front of the location, in particular in misdemeanor crimes.

Borough	Felony	Misdemeanor	Violation
Front of	548	886	483
Inside	1628	2889	1889
Opposite of	65	68	16
Rear of	29	44	15

Table 4.10: *Distribution of the explanatory variable Location of occurrence with respect to the three response categories.*

From Table 4.11 we can observe that the level *Brooklyn* has the majority of observations for all the three levels of the response. This means that in the Brooklyn borough there is in general an higher concentration of crimes. We can also observe that *Manhattan* and *Bronx* levels show a high frequency, in particular in correspondence of misdemeanor crimes.

In Table 4.12 we can observe that in case of felony crimes there is an higher frequency of attempted crimes with respect to the other two crime categories. The occurrence of non completed crimes with respect to the vi-

Borough	Felony	Misdemeanor	Violation
Bronx	509	1059	640
Brooklyn	697	1129	704
Manhattan	511	833	454
Queens	424	691	429
Staten Island	88	172	176

Table 4.11: *Distribution of the explanatory variable Borough name with respect to the three response categories.*

olation category is almost near to zero. This means that is more common that a violation crime is completed with respect to the other two categories.

Crime completed	Felony	Misdemeanor	Violation
Completed	2125	3866	2396
Not completed	104	18	7

Table 4.12: *Distribution of the explanatory variable Crimes completed with respect to the three response categories.*

The distribution of *Jurisdiction description* described in Table 4.13 shows that the most observed level is *N.Y. Police Dept.* in all the three crime categories.

Jurisdiction description	Felony	Misdemeanor	Violation
N.Y. Housing Police	258	477	251
N.Y. Police Dept.	1971	3407	2152

Table 4.13: *Distribution of the explanatory variable Jurisdiction description with respect to the three response categories.*

The distribution of *Suspect race* with respect to the three response categories is observable in Table 4.14. We observe that is more frequent that the suspect of a misdemeanor is a black person. The *unknown* level is the second most represented in the three crime categories.

In Table 4.15 we observe that the men commit the highest number of misdemeanor crimes and, in general, all the crime categories. There is a low frequency of women suspected of felony crimes.

Suspect race	Felony	Misdemeanor	Violation
Black	864	1431	960
Black Hispanic	267	210	140
Unknown	685	1027	432
White	173	316	389
White Hispanic	380	815	482

Table 4.14: *Distribution of the explanatory variable Suspect race with respect to the three response categories.*

Suspect sex	Felony	Misdemeanor	Violation
F	372	796	767
M	1243	2259	1407
U	614	829	229

Table 4.15: *Distribution of the explanatory variable Suspect sex with respect to the three response categories.*

The joint distribution of *Victim race* and the crime category is reported in Table 4.16. We observe that black people are the most frequent victims for all the crime categories followed by white Hispanic.

Victim race	Felony	Misdemeanor	Violation
Black	865	1558	966
Black Hispanic	123	252	150
Unknown	142	239	145
White	512	712	529
White Hispanic	587	1123	613

Table 4.16: *Distribution of the explanatory variable Victim race with respect to the three response categories.*

The distribution of *Victim sex* is described in Table 4.17 and shows that men victims are less frequent in all crime categories with respect to the women.

4.2 Results

The explanatory variables have been coded in a series of dummy variables for inclusion in the multinomial logistic regression model. Thereafter, the

Victim sex	Felony	Misdemeanor	Violation
Female	1246	2360	1618
Male	983	1524	785

Table 4.17: *Distribution of the explanatory variable Suspect sex with respect to the three response categories.*

dummy variables will be denoted in tables and figures with the following compact names:

- *Location of occurrence*:
  - Inside;
  - Opposite_Of;
  - Rare_Of;
- *Borough name*:
  - Brooklyn;
  - Manhattan;
  - Queens;
  - Staten Island;
- Suspect race:
  - Susp_Black_Hispanic;
  - Susp_Race_Unknown;
  - Susp_White;
  - Susp_White_Hispanic;
- *Suspect sex*:
  - Susp_Sex_M;
  - Susp_Sex_Unknown;
- *Victim race*:
  - Vict_Black_Hispanic;
  - Vict_Race_Unknown;
  - Vict_White;

  - Vict_White_Hispanic;
- *Victim sex*:
  - Vict_Sex_M;
- Crime_Complete;
- Juris_Police_Dept;

4.2.1 Log-Scores and Prediction Accuracy

As shown in Table 4.18 adaptive lasso has the best log-score followed by maximum likelihood. Ridge loses slightly in terms of log-score. Lasso log-score is better that ridge and similar to adaptive lasso and maximum likelihood. Principal components regression tends to perform worst than lasso, adaptive lasso and maximum likelihood, but it has a log-score better that ridge. Maximum likelihood low log-score value might be explained by the satisfactory number of observations contained in the train set. This allows to appreciate the asymptotic optimality of maximum likelihood estimation. Table 4.18 also contains the *baseline* log-score computed using the frequency of the three output categories in the train set. In other terms, the baseline log-score corresponds to the predictions obtained without using the information contained in the explanatory variables. The amount of information provided by the explanatory variables improve the performance of the fitting methods in terms of log-score.

Method	**Log-score**
Baseline log-score	3035.41
Maximum likelihood	2952.85
Ridge	2989.49
Lasso	2954.61
Adaptive Lasso	2951.17
Prin. Comp. Analysis	2975.14

Table 4.18: *Log-scores for the multinomial regression methods.*

Figure 4.2 reports the boxplots of the single components of the log-scores. This figure illustrates how maximum likelihood exposes to the risk of grossly incorrect predictions with respect to ridge, lasso and adaptive lasso that shrunk the predictions in this way reducing such risk. Principal components

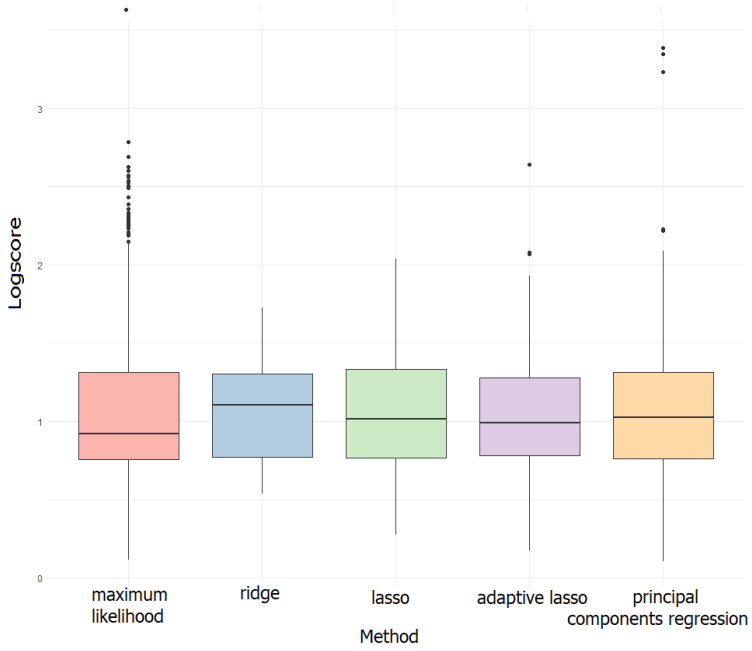

Figure 4.2: *Log-scores distributions for the fitting methods.*

regression is more exposed to the risk of incorrect predictions than ridge and the two lasso methods.
Table 4.19 shows that prediction accuracy is substantially equivalent for all methods, with maximum likelihood performing a little bit better than the other four methods. The prediction accuracy of the considered methods assume values between 45% and 47% and they outperform the baseline accuracy equal to 25%. The baseline accuracy prediction is referred to the predictions made without the information contained in the explanatory variables. The addition of the explanatory variables results into a strong improvement of the prediction accuracy with all the fitting methods.

The fact that methods that reduce the dimensionality perform comparably to maximum likelihood is an indication that a subset of the explanatory variables is sufficient.

4.2.2 Estimated Model Coefficients

Coefficients estimated with maximum likelihood, ridge, lasso and adaptive lasso are reported in Table 4.20. As for the simulation study, since the co-

Method	**Prediction accuracy %**
Baseline	25.40
Maximum likelihood	47.31
Ridge	46.21
Lasso	46.43
Adaptive Lasso	46.57
Prin. Comp. Analysis	45.79

Table 4.19: *Log-scores table for the multinomial regression methods.*

efficients estimated by the principal components regression have a different meaning from those computed with the other four methods, we have not reported them in these results, but we will describe them later on.
Table 4.20 reports the values of the estimates. The sign of the estimates denotes if the explanatory variable is positively or negatively associated to a crime category. Maximum likelihood does not perform the shrinkage on the coefficients and we observe that in general its coefficients have higher values with respect to the other methods. Maximum likelihood estimates indicate that it is more likely to observe a felony crime instead of a violation crime when the suspect is a man or the sex of the suspect is unknown, rather than a woman. Even in case of misdemeanor crimes instead of violation crimes it is more probable that the sex of the victim is man or unknown instead of woman. We observe also that it is less probable to observe a felony instead of a violation crime when the suspect is white or its ethnicity is unknown, rather than the suspect ethnicity is black. From the estimates of maximum likelihood we observe that it is more probable that a felony instead of a violation crime is committed in the borough of Brooklyn or Manhattan rather than in the Bronx. It is more probable that a misdemeanor crime instead of a violation crime is committed opposite of the location rather than in front of the location. Maximum likelihood estimates show that it is more likely that the crime is committed rear of the location instead of in front of the location in felony crimes instead of violation crimes. Then in felony crimes instead of violations crimes it is more probable that the victim is a man rather than a woman. Finally, it is more probable that the crime is not completed in felony crimes instead of violation crimes. What we have observed for maximum likelihood estimates holds also in ridge regression, lasso and adaptive lasso estimates.
Ridge assigns lower coefficients as effect of the shrinkage compared to maximum likelihood. Looking at lasso we observe the effect of the shrinkage on the coefficients. For example the explanatory variables *Rear of*, *Brooklyn*,

Manhattan and *Queens* are estimated to be exactly zero. Adaptive lasso assigns higher absolute values to those variables that also in ridge assume higher absolute values, and shrinks more the variables that are considered non-important by ridge. This is because the initial estimates of the coefficients in adaptive lasso are obtained through ridge regression. Adaptive lasso excludes the explanatory variables *Victim black Hispanic*, *Victim race unknown*, *Victim white* and *Victim white Hispanic*.

Var names	Maxmium likelihood cat 1	Maxmium likelihood cat 2	Ridge cat 1	Ridge cat 2	Lasso cat 1	Lasso cat 2	Adaptive Lasso cat 1	Adaptive Lasso cat2
Inside	-0.10	-0.07	-0.07	-0.05	0.00	0.00	0.00	0.00
Opposite_Of	0.10	0.78	0.05	0.25	0.01	0.11	0.01	0.33
Rear_Of	0.50	0.36	0.16	0.09	0.00	0.00	0.01	0.01
Brooklyn	0.31	0.10	0.04	0.01	0.00	0.00	0.00	0.00
Manhattan	0.34	0.11	0.06	0.02	0.00	0.00	0.00	0.00
Queens	0.20	0.00	0.01	-0.02	0.00	0.00	0.00	0.00
Staten Island	-0.26	-0.21	-0.15	-0.13	-0.09	-0.07	-0.23	-0.18
Susp_Black_Hispanic	-0.03	0.21	-0.02	0.05	0.00	0.00	0.00	0.00
Susp_Race_Unknown	-0.46	-0.26	0.12	0.09	0.00	0.00	0.00	0.00
Susp_White	-0.70	-0.48	-0.17	-0.18	-0.34	-0.38	-0.40	-0.47
Susp_White_Hispanic	-0.15	0.13	-0.02	0.04	0.00	0.00	0.00	0.00
Susp_Sex_M	0.72	0.48	0.02	0.03	0.22	0.18	0.17	0.14
Susp_Sex_Unknown	2.07	1.53	0.25	0.19	1.12	0.84	1.31	0.97
Vict_Black_Hispanic	-0.10	-0.06	-0.04	0.01	0.00	0.00	0.00	0.00
Vict_Race_Unknown	-0.11	-0.15	-0.02	-0.02	0.00	0.00	0.00	0.00
Vict_White	0.12	-0.14	0.01	-0.07	0.01	-0.06	0.00	0.00
Vict_White_Hispanic	0.19	0.11	0.03	0.05	0.00	0.00	0.00	0.00
Vict_Sex_M	0.42	0.31	0.11	0.08	0.19	0.14	0.22	0.16
Crime_Complete	-2.77	-0.67	-0.62	0.03	-1.64	-0.08	-2.65	-0.56
Juris_Police_Dept	-0.09	-0.15	-0.02	-0.05	0.00	0.00	0.00	0.00

Table 4.20: *Table of the estimated coefficients. We highlighted the most important coefficients in red.*

4.2.3 Principal Components Regression Analysis

Figure 4.3 shows the portion of the cumulative standard deviation explained by the principal components of the design matrix. We decided to consider the principal components that cumulate the 70% of the standard deviation. With respect to the initial 20 explanatory variables, the number of selected principal components is twelve corresponding to a good dimensionality reduction. The first four principal components explain only the 25% of the cumulative standard deviation of the data. Instead, if we consider the first eight principal components the portion of cumulative standard deviation explained becomes 50%.

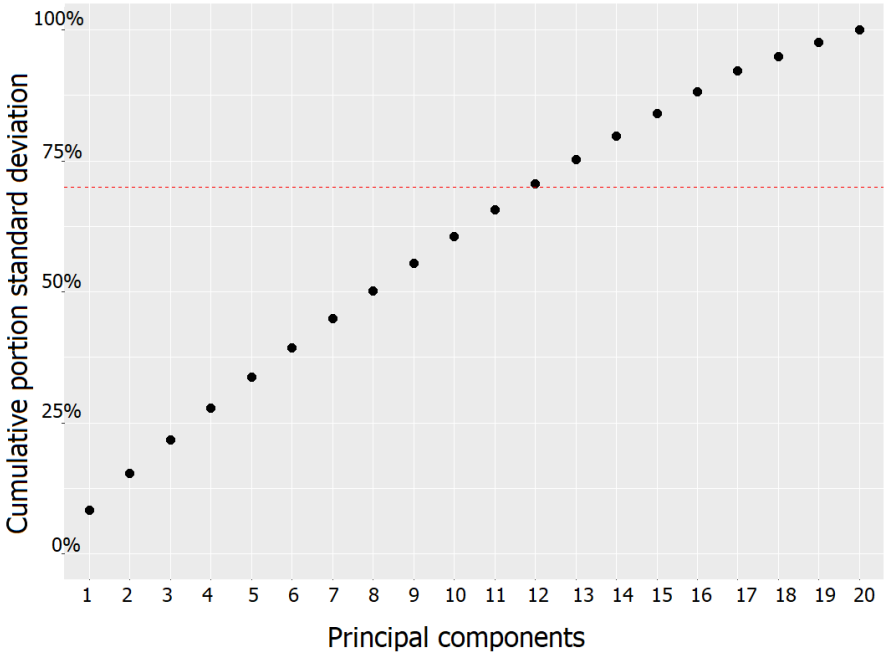

Figure 4.3: *Principal components for the principal components regression.*

Table 4.21 shows the loadings of the explanatory variables. We consider important the correlations that are above 40%. The first principal component is negatively correlated to a black suspect and positively to unknown suspect race. There is a positive correlation with a man suspect but negative when the sex of the suspect is unknown. The second principal component is negatively correlated with white suspect and to white victim. The third principal component is negatively correlated with the borough Queens. The fourth principal component is highly negatively correlated to the borough Brooklyn and positively correlated with the Manhattan borough. The fifth principal component is negatively correlated to the location inside and positively correlated with black Hispanic suspect and black Hispanic victim. The sixth principal component has negative correlation with the borough Manhattan but positive with the borough Queens. The seventh principal component is highly positively correlated to victim ethnicity unknown. The eighth principal component has a negative correlation with the borough Manhattan and positive to the borough Queens. It is also negatively correlated to crime complete. The ninth component is strongly positively correlated to the borough Staten Island. The tenth principal component has positive correlation with a male victim and with the police department jurisdiction. The eleventh

principal component is positively correlated to location opposite of, and negatively to the location rear of. We observe also a strong negative correlation with crime complete. The last principal component is negatively correlated to black Hispanic suspects, and it is positively correlated to white Hispanic suspects and black Hispanic victims.
From the loadings emerges that the first principal component denotes that the suspect ethnicity is not unknown and the suspect sex is man and not unknown. The second principal component denotes that the victim and the suspect are not white and that the victim is white Hispanic. The third principal component denotes that the borough is not Queens. The fourth principal component denotes that the borough is Manhattan. The seventh principal component denotes that the victim sex is unknown. The ninth principal component denotes that the crime occurs in the borough Staten Island. The eleventh principal components denotes that the crime occurs opposite of the location and not rare of the location and that the crime has not been completed. The last principal component denotes that the crime is not committed by a black Hispanic suspect and that the suspect is white. The last principal components denotes also that the victim is black Hispanic.

Explanatory Vars	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Inside	0.159	-0.048	0.240	0.183	-0.467	0.300	0.093	0.085	0.006	0.125	0.026	0.036
Opposite_Of	-0.114	0.050	-0.098	-0.170	0.246	-0.152	-0.212	-0.471	-0.079	-0.348	0.500	0.070
Rear_Of	-0.033	0.019	-0.130	-0.077	0.358	-0.285	0.136	0.487	0.058	-0.269	-0.410	-0.048
Brooklyn	-0.013	-0.107	0.215	-0.711	-0.133	0.005	-0.093	0.036	-0.122	0.118	-0.088	-0.047
Manhattan	-0.117	-0.021	0.208	0.544	-0.113	-0.478	0.036	-0.120	-0.265	0.002	-0.015	-0.046
Queens	0.029	0.067	-0.446	0.145	0.166	0.481	0.341	0.124	-0.158	-0.222	0.144	0.055
Staten Island	0.021	-0.234	-0.149	0.101	-0.005	-0.061	-0.269	-0.023	0.814	0.008	-0.004	0.069
Susp_Black_Hispanic	0.119	0.088	0.320	0.189	0.396	0.249	-0.170	-0.026	0.083	0.081	0.010	-0.565
Susp_Race_Unknown	-0.532	0.130	-0.016	0.031	-0.083	0.070	-0.023	0.064	0.039	-0.012	-0.003	-0.021
Susp_White	0.068	-0.542	-0.162	0.081	0.000	0.008	-0.061	-0.035	-0.041	-0.066	-0.014	-0.065
Susp_White_Hispanic	0.274	0.298	-0.307	0.028	-0.144	-0.173	-0.080	-0.040	0.001	0.200	-0.068	0.412
Susp_Sex_M	0.438	-0.115	-0.007	-0.030	0.139	-0.141	0.090	-0.066	-0.123	-0.024	0.036	-0.059
Susp_Sex_Unknown	-0.548	0.136	-0.028	0.027	-0.082	0.075	-0.064	0.069	0.043	0.003	-0.010	0.008
Vict_Black_Hispanic	0.034	0.091	0.340	0.163	0.424	0.222	-0.213	0.077	0.010	0.150	-0.002	0.576
Vict_Race_Unknown	-0.060	0.027	0.099	-0.038	0.058	-0.111	0.715	-0.362	0.352	0.103	-0.021	-0.020
Vict_White	-0.137	-0.533	-0.157	0.132	-0.054	-0.017	-0.128	0.088	-0.176	0.015	0.011	0.047
Vict_White_Hispanic	0.147	0.418	-0.333	0.072	-0.160	-0.095	-0.291	0.006	0.031	0.074	-0.039	-0.358
Vict_Sex_M	-0.156	-0.040	-0.132	-0.029	0.310	-0.186	0.009	-0.144	-0.159	0.489	-0.231	0.023
Crime_Complete	0.013	0.021	0.024	0.043	-0.080	0.232	-0.141	-0.511	-0.066	-0.393	-0.696	0.074
Juris_Police_Dept	-0.083	-0.118	-0.320	-0.001	0.106	0.245	-0.022	-0.242	-0.075	0.493	-0.081	-0.119

Table 4.21: *Loadings of the principal components.*

Conclusions

In the real data application and in the simulation study, we applied five fitting methods for multinomial regression that are characterized by different behaviour and performance. Maximum likelihood does not provide dimensionality reduction, suffers for asymmetry and sparsity of explanatory variables. For this reason, maximum likelihood exposes to the risk of grossly incorrect predictions. Maximum likelihood performances decreases as the correlation between the explanatory variables raises. As saw in the simulation study, maximum likelihood provides better predictions if there is a substantial sample dimension.
Ridge regression penalizes the explanatory variables coefficients shrinking them to zero. Lasso performs a selection of the explanatory variables coefficients since non-important coefficients can be estimated to be exactly zero so that the corresponding explanatory variables are excluded from the analysis. Adaptive lasso shrinks the non-important coefficients to zero, starting from initial estimates obtained by ridge regression. Therefore, adaptive lasso performs a different penalization on each coefficient with respect to lasso, because it aims to preserve important explanatory variables. Ridge, lasso and adaptive lasso performances are less influenced by the sample size with respect to maximum likelihood as shown in the simulation study. In the real data application ridge tends to shrinks the non-important explanatory variables towards zero, while lasso and adaptive lasso select some explanatory variables and shrinks to zero some others. Adaptive lasso consider important the explanatory variables that also for ridge are important and shrinks to zero or assigns very little importance to the explanatory variables that are also non-important in ridge.
Principal components regression reduces the dimensionality through the principal components analysis of the original explanatory variables. These principal components are used as explanatory variables in the multinomial logistic regression model. Principal components regression is more sensible to the sample size compared to the other three dimensionality reduction methods. Principal components regression performance when the correlation between

the explanatory variables raises is somehow comparable to ridge, lasso and adaptive lasso. We obtained a good dimensionality reduction in the New York City crimes data.
In conclusion, dimensionality reduction methods improves fitting of multinomial regression methods because they provide better control of the correlation between the explanatory variables, asymmetry and sparsity with respect to traditional maximum likelihood estimation. These desirable properties of dimensionality reduction methods translate in improved predictions.

Appendix

R Code

```
## Libraries imports
library(Hmisc)
library(MASS)
library(ISLR)
library(glmnet)
library(pls)
library(gridExtra)
library(grid)
library(histogram)
library(ggplot2)
library(reshape2)
library(nnet)
library(RColorBrewer)
library(data.table)
library(gtools)


## APPLICATION FUNCTIONS
print.logscore.matrix <- function(res){
  ## Print the distributions of the log-scores for the
     real data application

  ## Logscore matrices boxplots
  ml.logscore <- -sum(res$logscore$ml$logscore)
  ridge.logscore <- -sum(res$logscore$ridge$logscore)
  lasso.logscore <- -sum(res$logscore$gr.lasso$logscore
     )
  alasso.logscore <- -sum(res$logscore$gr.alasso$
     logscore)
  pcr.logscore <- -sum(res$logscore$pcr.reg$logscore)
  coefs <- data.table(ml = res$logscore$ml$logscore,
```

```
  ridge = res$logscore$ridge$logscore,
  lasso = res$logscore$gr.lasso$logscore,
  alasso = res$logscore$gr.alasso$logscore,
  pcr = res$logscore$pcr.reg$logscore)

  coefs[, name := c("ml", "ridge", "lasso", "alasso", "
     pcr")]
  to_plot = melt(coefs, id.vars = "name", variable.name
      = "method", value.name = "logscore")
  myplot <- ggplot(to_plot, aes(x=method, y=logscore,
     fill = method)) + geom_boxplot() + scale_fill_
     brewer(palette="Pastel1") + theme_minimal()+ xlab(
     "Method") + ylab("Logscore")+ ggtitle("Log-scores
     matrices for the 5 methods")
  print(myplot)
}

print.tab.coefs <- function(res){
  ## Print regression coefficients (ml, ridge, lasso,
     adaptive lasso) as table

  ml <- (round(data.frame(res$coefficients$ml$coefs),
     2))
  ridge <- (round(data.frame(res$coefficients$ridge$
     coefs), 2))
  lasso <- (round(data.frame(res$coefficients$gr.lasso$
     coefs), 2))
  alasso <- (round(data.frame(res$coefficients$gr.
     alasso$coefs), 2))
  varnames <- c("Inside", "Opposite_Of", "Rear_Of", "
     Brooklyn", "Manhattan", "Queens", "Staten Island",
      "Susp_Black_Hispanic", "Susp_Race_Unknown", "Susp
     _White", "Susp_White_Hispanic", "Susp_Sex_M", "
     Susp_Sex_Unknown", "Vict_Black_Hispanic", "Vict_
     Race_Unknown", "Vict_White", "Vict_White_Hispanic"
     , "Vict_Sex_M", "Crime_Complete", "Juris_Police_
     Dept")
  vars.coefs <- cbind(matrix(varnames, ncol = 1, nrow =
      length(varnames)), ml, ridge, lasso, alasso)
  png("Coefficients_regr_table.png", width = 1024,
     height = 728, units = "px")
  colnames(vars.coefs) <- c("Var names", "Maxmium
     likelihood\ncat 1", "Maxmium likelihood\ncat 2", "
```

```
      Ridge\ncat 1", "Ridge\ncat 2", "Lasso\ncat 1", "
      Lasso\ncat 2", "Adaptive Lasso\ncat 1", "Adaptive
      Lasso\ncat2")
  grid.table(vars.coefs, rows = NULL)
  dev.off()
}
print.sdevs <- function(pc){
  ## Print the cumulative standard deviations of the
     computed principal components

  pc_sdevs <- cumsum(pc$sdev)/sum(pc$sdev)
  tot_prcomp <- length(pc$sdev)
  vect_n <- "PC01"
  for (i in 2:tot_prcomp){
    vect_n <- append(vect_n, paste("PC0", i, sep=''))
  }
  sdev <- data.frame(prcomp = as.factor(vect_n), sdev =
      round(pc_sdevs*100, 3))
  png("PCR_sdevs.png", width = 1024, height = 728,
     units = "px")
  myplot <- ggplot(sdev, aes(x=reorder(prcomp, sdev,
     mean), y=sdev)) + geom_point(size = 5) + ylim(c(0,
      100)) + geom_hline(yintercept=70, linetype="
     dashed", color = "red") + xlab("Principal
     components") + ylab("Cumulative portion standard
     deviation")
  print(myplot)
  dev.off()
}
print.loadings <- function(pc, selected.comp){
  ## Print the squared loadings for the principal
     components regression

  load <- round(data.frame(loadings(pc)[,1:selected.
     comp]), 3)
  varnames <- c("Inside", "Opposite_Of", "Rear_Of", "
     Brooklyn", "Manhattan", "Queens", "Staten Island",
      "Susp_Black_Hispanic", "Susp_Race_Unknown", "Susp
     _White", "Susp_White_Hispanic", "Susp_Sex_M", "
     Susp_Sex_Unknown", "Vict_Black_Hispanic", "Vict_
     Race_Unknown", "Vict_White", "Vict_White_Hispanic"
     , "Vict_Sex_M", "Crime_Complete", "Juris_Police_
     Dept")
```

```
  load <- cbind(matrix(varnames, ncol = 1, nrow =
     length(varnames)), load)
  vect_n <- "Explanatory Vars"
  for (i in 1:selected.comp){
    vect_n <- append(vect_n, paste("PC", i, sep=''))
  }
  colnames(load) <- vect_n
  png("loadings_pcr.png", width = 1024, height = 728,
     units = "px")
  grid.table(load, rows = NULL)
  dev.off()
}


## SIMULATION FUNCTIONS
aggreg <- function(x, method, n.var, n.categ, trials) {
  ## Auxiliary function to make the coefficients easier
      to handle

  res <- array(NA, dim = c(n.var, n.categ, trials))
  for (i in seq_len(trials))
  res[,,i] <- as.numeric(x[2,][[i]][[method]][[1]])
  res
}
print.percpred <- function(res, trials){
  ## Print the missclassification error for the
     regression methods
  vect <- matrix(as.numeric(unlist(res[4,])), byrow=T,
     nrow=ncol(res))
  vect <- vect[complete.cases(vect),]
  miss.vect <- colSums(vect)/dim(vect)[1]
  misserrors <- data.frame(method = c("ml", "ridge", "
     lasso", "ad. lasso", "pcr"), misserr = round(miss.
     vect,3))
  png("perc_pred.png", width = 1024, height = 728,
     units = "px")
  myplot <- ggplot(misserrors, aes(x=reorder(method,
     misserr, mean), y=misserr, color = method)) + geom
     _text(aes(label=misserr), vjust=2, color="black",
     size=6) + geom_point(size = 6) + ylim(c(0, 100))+
      scale_fill_brewer(palette="Pastel1") + theme_
     minimal()+ xlab("Method") + ylab("Prediction
     accuracy %")+ ggtitle("Prediction accuracy %")
```

```
  print(myplot)
  dev.off()
}
print.boxplots <- function(logscores, corr){
  ## Print a paired-boxplot for log-score analysis

  colors <- c("lightgoldenrodyellow", "lightsteelblue3"
     , "mistyrose", "palegreen", "lightsalmon", "
     antiquewhite1")
  title <- paste("Log-score distribution\n at corr=",
     corr, sep = '')
  methods <- c("Maximum Likelihood", "Ridge", "Lasso",
     "Adaptive Lasso", "PCR")
  filename <- paste("boxplot_methods", corr, ".png",
     sep='')
  png(filename, width = 1024, height = 728, units = "px
     ")
  boxplot(matrix(as.numeric(unlist(logscores[1,])),
     byrow=T, nrow=ncol(logscores)), col = colors, main
      = title, names = methods)
  dev.off()
  }
  print.lasso.vars <- function(coefs, type){
  ## print selected and unselected variables for lasso
  colnames(coefs) <-  c("Method", "Categ 1\ncorrect
     selection (%)", "Categ 2\ncorrect selection (%)")
  filename <- paste("lasso_select_vars",type, ".png",
     sep='')
  png(filename, width = 1024, height = 728, units = "px
     ")
  grid.table(coefs)
  dev.off()
}
lasso.vars.analysis <- function(coefs, n.var, ncateg,
   trials){
  ## Check the selected and unselected variables by
     lasso methods

  coefslasso <-  aggreg(coefs, "gr.lasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
  coefsalasso <- aggreg(coefs, "gr.alasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
```

```
  imp.lasso <- cbind("Lasso", 100 * mean(coefslasso
     [1:10,1,]!=0), 100 * mean(coefslasso[1:10,2,]!=0))
  imp.alasso <- cbind("Ad. Lasso", 100 * mean(
     coefsalasso[1:10,1,]!=0), 100 * mean(coefsalasso
     [1:10,2,]!=0))
  poor.lasso <- cbind("Lasso", 100 * mean(coefslasso
     [11:20,1,]==0), 100 * mean(coefslasso
     [11:20,2,]==0))
  poor.alasso <- cbind("Ad. Lasso", 100 * mean(
     coefsalasso[11:20,1,]==0), 100 * mean(coefsalasso
     [11:20,2,]==0))
  print.lasso.vars(rbind(imp.lasso, imp.alasso), "
     Important")
  print.lasso.vars(rbind(poor.lasso, poor.alasso), "
     Poor")
}
print.coefs <- function(vect1, vect2, vect3, vect4, id.
   coefs, cat, m1, m2, m3, m4, val){
  ## Print the distribution of the extracted
     coefficients using a paired-boxplot

  colors <- c("lightgoldenrodyellow", "lightsteelblue3"
     , "mistyrose", "palegreen", "antiquewhite1")
  title <- paste("Boxplot of coefficient", id.coefs, "
     for cat", cat)
  filename <-paste("Boxplot_coefficient_", id.coefs,"_"
     , cat,".png", sep = '')
  idx <- seq(1, length(vect1))
  x.frame<- data.frame(idx, vect1, vect2, vect3, vect4)
  data <- melt(x.frame, id.vars = "idx")
  png(filename, width = 1024, height = 728, units = "px
     ")
  boxplot(vect1, vect2, vect3, vect4, col = colors,
     main = title, names = c(m1,m2,m3,m4))
  abline(h=val)
  dev.off()
}
analyse.coeffs <- function(coefs, n.var, ncateg, trials
   ){
  ## Extract the coefficients for all methods (no pcr)

  coefsml<- aggreg(coefs, "ml", n.var = n.var, n.categ
     = ncateg-1, trials = trials)
```

```
  coefsridge <- aggreg(coefs, "ridge", n.var = n.var, n
     .categ = ncateg-1, trials = trials)
  coefslasso <-  aggreg(coefs, "gr.lasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
  coefsalasso <- aggreg(coefs, "gr.alasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
  categ <- c(1, 2)
  id.coefs<- c(1, 9, 16)
  for (j in id.coefs){
    for(i in categ){
      if (j<5){
        val <-3
      }
      else{
        if (j>5 && j<10)
        val <-1
        else val <-0
      }
        print.coefs(coefsml[j, i,], coefsridge[j, i,],
           coefslasso[j, i,], coefsalasso[j, i,], j, i,
            "ML", "Ridge", "Lasso", "Adaptive-Lasso",
           val)
      }
    }
  }
  sqrt.mse <- function (val, real.val){
  ## Compute the squared mse of the regression methods

  bias <- mean(val, na.rm = T) - real.val
  sqrtmse <- sqrt(bias^2 + var(val, na.rm = T))
  return(sqrtmse)
}
print.mse <- function(mse, method){
  ## Auxiliary function to print the root mse

  colnames(mse) <- c("# variable", "Categ 1","Categ 2")
  filename <- paste("sqrtmse", method, ".png", sep='')
  png(filename, width = 1024, height = 728, units = "px
     ")
  grid.table(mse)
  dev.off()
}
```

```
analyse.mse <- function(coefs, n.var, ncateg, trials,
   beta){
  ## Extract the coefficients and compute the root mse
     for all methods

  coefsml<- aggreg(coefs, "ml", n.var = n.var, n.categ
     = ncateg-1, trials = trials)
  coefsridge <- aggreg(coefs, "ridge", n.var = n.var, n
     .categ = ncateg-1, trials = trials)
  coefslasso <-  aggreg(coefs, "gr.lasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
  coefsalasso <- aggreg(coefs, "gr.alasso", n.var = n.
     var, n.categ = ncateg-1, trials = trials)
  ml <- matrix(rep(1:n.var), n.var, ncateg)
  ridge <- matrix(rep(1:n.var), n.var, ncateg)
  lasso <- matrix(rep(1:n.var), n.var, ncateg)
  alasso <- matrix(rep(1:n.var), n.var, ncateg)
  for (j in 1:n.var){
    for (i in 1:(ncateg-1)){
      ml[j,i+1] <- round(sqrt.mse(coefsml[j,i,], beta[j
         ]),3)
      ridge[j,i+1] <- round(sqrt.mse(coefsridge[j,i,],
         beta[j]), 3)
      lasso[j,i+1] <- round(sqrt.mse(coefslasso[j,i,],
         beta[j]), 3)
      alasso[j,i+1] <- round(sqrt.mse(coefsalasso[j,i
         ,], beta[j]),3)
    }
  }
  print.mse(ml, "ML")
  print.mse(ridge, "Ridge")
  print.mse(lasso, "Lasso")
  print.mse(alasso, "Adaptive-Lasso")
}
print.sel.comp <- function(pr.comp, n.var, imp.var){
  ## Plot the selected principal components

  png("Selected_principal_components.png", width =
     1024, height = 728, units = "px")
  plot(as.numeric(pr.comp[3,]), type = "l", ylim = c(5,
     n.var), lwd = 2, main = "Number of selected\
     nprincipal components", col = "coral1", xlab = "
     Trails", ylab = "Prin comp")
```

```
  dev.off()
}
red.regr <- function(X, y, method = c("ML", "Ridge", "
   Lasso", "ALasso"), type = "grouped", train = NULL,
   test = NULL){
  ## Perform the regression for the multinomial
     logistic model

  method <- match.arg(method)
  type <- match.arg(type)
  id.lasso <- if (method == "Lasso") 1 else 0
  if(is.null(train)){
    ##split the dataset into train (80%) and test (20%)
        set
    train <- sample(1:nrow(X), 0.8 * nrow(X))
    test <- (-train)
  }
  lambda <- if (method == "ML") sqrt(.Machine$double.
     eps) else NULL
  mod <- glmnet(X[train, ], y[train], alpha = id.lasso,
      lambda = lambda, family = "multinomial", type.
     multinomial = type)
  if (method != "ML") {
    mod.cv <- cv.glmnet(X[train, ], y[train], alpha =
       id.lasso, lambda = lambda, family="multinomial",
        type.measure = "class")
  }
  mod.bestlam <- if (method == "ML") sqrt(.Machine$
     double.eps) else mod.cv$lambda.1se
  mod.pred <- predict(mod, s = mod.bestlam, newx = X[
     test, ], type = "response")
  coefs <- coef(mod, s = mod.bestlam)

  if(method == "ALasso"){
    best.coef <- do.call(cbind, coef(mod.cv, s = mod.
       bestlam))
    best.weights <- 1 / abs(as.matrix(best.coef)[-1,])
    model <- glmnet(X[train, ], y[train], alpha = 1,
       penalty.factor = best.weights, family = "
       multinomial", type.multinomial = type)
    model.cv <- cv.glmnet(X[train, ], y[train], alpha =
        1, penalty.factor = best.weights, family = "
       multinomial", type.measure = "class", keep =
```

```
      TRUE)
    model.bestlam <- model.cv$lambda.1se
    mod.pred <- predict(model, s = model.bestlam, newx
       = X[test, ], type = "response", parallel = T)
    coefs <- coef(model, s = model.bestlam)
  }
  test.preds <- nrow(mod.pred)
  for (i in 1:nrow(mod.pred))
  test.preds[i] <- mod.pred[i, y[test][i], ]
  logscore <- -sum(log(test.preds))
  ok_class <- (sum(max.col(mod.pred[,1:3,1])==y[test])/
     length(y[test]))*100
  res <- list(logscore = logscore, coefs = matrix(
     unlist(lapply(coefs, function(x) as.numeric(x))),
     byrow=F, ncol=3), miss.err = ok_class)
  return(res)
}

### RUN SIMULATION
simone <- function(ncateg, nobs, nvar, rho, beta, alpha
   ) {
  ## Function used to simulate once a given scenario

  ## Compute the correlation matrix (no equi-
     correlation)
  sigma <- toeplitz(rho ^ seq(from = 0, to = nvar - 1))
  ## Compute the design matrix X
  X <- mvrnorm(nobs, mu = rep(0.0, nvar), Sigma = sigma
     )
  exp.terms <- probs <- matrix(0.0, nrow = nobs, ncol =
      ncateg)
  for (j in 1:(ncateg-1))
  exp.terms[,j] <- exp(alpha[j] + X %*% beta[,j])
  ## Compute the probability matrix
  probs <- exp.terms / (1 + rowSums(exp.terms))
  probs[, ncateg] <- 1 / (1 + rowSums(exp.terms))
  ## Simulate the responses
  y <- rMultinom(probs, m=1)

  ## Perform principal components analysis
  pc <- prcomp(X, scale. = T)
  ## Select the number of principal components that
     exceeded the inferior limit
```

```
selected.comp <- min(which(cumsum((pc$sdev)/sum(pc$
   sdev))>0.7))
X.pc <- pc$x[,seq_len(selected.comp)]

compute.coefs <- function(x) {
  ## Function that change the Poisson likelihood
     coeffcients
  ## Into maximul likelihood.
  ## This function delete also the intercept alpha

  x <- x[-1, ]
  x[, (1:ncateg-1)] - x[, ncateg]
}
nonconv <- function(x) {
## Auxiliary function used in case of try-error

  x <- list()
  x$logscore <- NA
  x$coefs <- matrix(NA, nrow = nvar, ncol = ncateg -
     1)
  X$miss.err <- NA
}

## Maximul likelihood
ml <- try(red.regr(X, y, "ML", "grouped"), silent =
   TRUE)
if (class(ml) == "try-error") nonconv(ml)
else ml$coefs <- compute.coefs(ml$coefs)

## Ridge regression
ridge <- try(red.regr(X, y, "Ridge", "grouped"),
   silent = TRUE)
if (class(ridge) == "try-error") nonconv(ridge)
else ridge$coefs <- compute.coefs(ridge$coefs)

## The lasso
gr.lasso <- try(red.regr(X, y, "Lasso", "grouped"),
   silent = TRUE)
if (class(gr.lasso) == "try-error") nonconv(gr.lasso)
else gr.lasso$coefs <- compute.coefs(gr.lasso$coefs)

## The adaptive lasso
```

```
  gr.alasso <- try(red.regr(X, y, "ALasso", "grouped"),
      silent = TRUE)
  if (class(gr.alasso) == "try-error") nonconv(gr.
     alasso)
  else gr.alasso$coefs <- compute.coefs(gr.alasso$coefs
     )

  ## Principal components regression applied to Maximum
      likelihood
  pcr <- try(red.regr(X.pc, y, "ML", "grouped"), silent
      = TRUE)
  if (class(pcr) == "try-error") nonconv(pcr)
  else pcr$coefs <- compute.coefs(pcr$coefs)

  all.methods <- list(ml, ridge, gr.lasso, gr.alasso,
     pcr)
  logscore <- unlist(lapply(all.methods, function(x) x[
     "logscore"]))
  names(logscore) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")
  coefs <- lapply(all.methods, function(x) x["coefs"])
  names(coefs) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")
  all.methods <- list(ml, ridge, gr.lasso, gr.alasso,
     pcr)
  misserr <- unlist(lapply(all.methods, function(x) x["
     miss.err"]))
  names(misserr) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")

  list(logscore = logscore, coefficients = coefs,
     princomp = selected.comp, misserr = misserr)
}

### RUN APPLICATION
apply.regr <- function(X, y, train, test) {
  ## Function that applies the regression methods on
     real data

  ## Number of output categories
  ncateg <- length(unique(y))
  ## Number of explanatory variables
  nvar <- ncol(X)
```

```
## Perform principal components analysis
pc <- prcomp(X, scale. = T)
## Select the number of principal components that
   exceeded the inferior limit
selected.comp <- min(which(cumsum(pc$sdev)/sum(pc$
   sdev)>0.7))
X.pc <- pc$x[,seq_len(selected.comp)]

## Print loadings for the computed principal
   components
print.loadings(pc, selected.comp)
print.sdevs(pc)
compute.coefs <- function(x) {
  ## Function that change the Poisson likelihood
     coeffcients
  ## Into maximul likelihood.
  ## This functiondelete also the intercept alpha

  x <- x[-1, ]
  x[, (1:ncateg-1)] - x[, ncateg]
}
nonconv <- function(x) {
  ## Auxiliary function used in case of try-error

  x <- list()
  x$logscore <- NA
  x$coefs <- matrix(NA, nrow = nvar, ncol = ncateg -
     1)
}

## Maximul likelihood
ml <- try(red.regr(X, y, "ML", "grouped", train, test
   ), silent = TRUE)
if (class(ml) == "try-error") nonconv(ml)
else ml$coefs <- compute.coefs(ml$coefs)

## Ridge regression
ridge <- try(red.regr(X, y, "Ridge", "grouped", train
   , test), silent = TRUE)
if (class(ridge) == "try-error") nonconv(ridge)
else ridge$coefs <- compute.coefs(ridge$coefs)
```

```
  ## The lasso
  gr.lasso <- try(red.regr(X, y, "Lasso", "grouped",
     train, test), silent = TRUE)
  if (class(gr.lasso) == "try-error") nonconv(gr.lasso)
  else gr.lasso$coefs <- compute.coefs(gr.lasso$coefs)

  ## The adaptive lasso
  gr.alasso <- try(red.regr(X, y, "ALasso", "grouped",
     train, test), silent = TRUE)
  if (class(gr.alasso) == "try-error") nonconv(gr.
     alasso)
  else gr.alasso$coefs <- compute.coefs(gr.alasso$coefs
     )

  ## Principal components regression applied to Maximum
      likelihood
  pcr <- try(red.regr(X.pc, y, "ML", "grouped", train,
     test), silent = TRUE)
  if (class(pcr) == "try-error") nonconv(pcr)
  else pcr$coefs <- compute.coefs(pcr$coefs)

  all.methods <- list(ml, ridge, gr.lasso, gr.alasso,
     pcr)
  logscore <- unlist(lapply(all.methods, function(x) x[
     "logscore"]))
  names(logscore) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")
  coefs <- lapply(all.methods, function(x) x["coefs"])
  names(coefs) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")
  all.methods <- list(ml, ridge, gr.lasso, gr.alasso,
     pcr)
  misserr <- unlist(lapply(all.methods, function(x) x["
     miss.err"]))
  names(misserr) <- c("ml", "ridge", "gr.lasso", "gr.
     alasso", "pcr.reg")

  list(logscore = logscore, coefficients = coefs,
     princomp = selected.comp, misserr = misserr)
}
```

```
## Importing the R function file
source("SimulationApplicationFunctions.R")
```

```
set.seed(1024)
## Number of explanatory variables
n.var <- 20
## Number of explanatory variables with values
   different from zero
imp.var <- 10
## Number of observation of the design matrix X
n.obs <- 500
## Number of simulation repetitions
trials <- 500
## Number of output categories
ncateg <- 3
## correlation among the explanatory variables
rho <- c(0, 0.3, 0.6, 0.9)
## Coefficients vector
betas <- matrix(0.0, nrow = n.var, ncol = ncateg)
betas[1:round(imp.var/2), 1:(ncateg - 1)] <- 3
betas[(round(imp.var/2)+1):imp.var, 1:(ncateg - 1)] <-
   1
## Intercept vector
alpha <- c(-.5, .5, 0)

## start simulation
sim <- replicate(trials, (simone(ncateg, n.obs, n.var,
   rho, betas, alpha)))
## Print the accuracy prediction
print.percpred(sim, trials)
## Plot the logscore distributions
print.boxplots(sim, rho)
## Print the accuracy on selecting explanatory
   variables (lasso)
lasso.vars.analysis(sim, n.var, ncateg, trials)
## Analysis of the coefficients distributions (1, 9,
   16)
analyse.coeffs(sim, n.var, ncateg, trials)
## Print root mse for the coefficients
analyse.mse(sim, n.var, ncateg, trials, betas)
## Print selected principal components
print.sel.comp(sim, n.var, imp.var)
```

```
## Importing the R function file
source("SimulationApplicationFunctions.R")
set.seed(2024)
```

```
## Import the dataset
my.data <- read.csv2("pol.csv")
dat <- data.frame(my.data)
## Set the design matrix
X <- model.matrix(~LOC_OF_OCCUR_DESC + BORO_NM + SUSP_
   RACE + SUSP_SEX + VIC_RACE + VIC_SEX + CRM_ATPT_CPTD
   _CD + JURIS_DESC, data = dat)[,-1]
## Set the output vector
Y <- as.numeric(dat$LAW_CAT_CD)
## The train set includes February and March
   observations
train <- 1:5677
## Predictions are made on April observations
test <- 5678:8516
## Compute the reference log-score
prop <- prop.table(table(Y[train]))
baseline_logscore <- -sum(log(prop[Y[test]]))
baseline_Accuracy <- sum(max.col(prop[Y[test]])==Y[test
   ])/length(Y[test])*100
## RUN APPLICATION
res <- apply.regr(X, Y, train, test)

## Save the obtained results
print.accuracy(res)
print.tab.accuracy(res)
print.logscore.matrix(res)
print.logscores(res, ref_logscore)
print.tab.coefs(res)
```

Bibliography

Agresti, A. (2003). *Categorical Data Analysis*. Wiley Series in Probability and Statistics, 2nd edition.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, 2nd edition.

NYCOpenData (2018). `https://opendata.cityofnewyork.us/` New York City Open Data Website.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rodríguez, G. (2007). Lecture notes on generalized linear models.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Acknowledgements

The end of these five years brings me to thought about all the people that took part to this journey, someone from the beginning and someone else along the road. It is the end of a trip, from which I could learn a lot. The people that I met in these years changed me, and I hope in some way they make me a better human being. Firstly, I want to thank my family. Thanks to my parents Franca and Sandro. They made possible this experience and they shared with me all the moments, the good and the bad ones. They always gave to me all that I needed and much more, with hard work and sacrificing a lot to allow to me to obtain the graduation. I owe them so much, and I hope they could be always proud of me. Thanks to Luca that in this last year and a half has been fundamental to me. He always encouraged me to give all of myself in what I was doing and it has been thanks to his support and staying next to me if I could get through the indecision moments, but at the same time, if I could reach with much more determination all of my objectives. Thanks to my brother Saverio that with his affection and attitude always relieves my mood. Thanks to Martina, Daniele, Marco, Diletta and Riccardo who are the best friends and classmates that I always wished to have. Thanks to Paolo that is really hard to leave after five years of cohabitation. Thanks to Marica, Milena, Melania, Fabio and Emanuele that from home cheered for me. Thanks to all of my professors that in these five years leaded me with success through this path with great competence and availability. Thanks to my supervisor, Ch. Prof. Cristiano Varin who, in these last months, has been always really available and ready to satisfy all my requests and also the insistences. Thanks to his experience, patience and supervision this thesis could have see the end. Finally, thank to all the people that I do not mention but who contribute with their help and their friendship to make these five years one of the best experiences of my life.