Shrinkage Estimators under Generalized Garrote and LINEX Loss Functions for Regression Analysis

by

Munaweera Arachchilage Inesh Prabuddha

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

Master of Science

Department of Statistics
University of Manitoba
Winnipeg

Abstract

Shrinkage methods are widely used in multiple linear regression analysis to address the multicollinearity and some other issues in many practical situations. Most of the commonly used shrinkage methods such as ridge regression and lasso do not consider the importance of each variable when applying the shrinkage, and all model coefficients will be shrunken towards zero in a similar rate. In other words, those methods shrink least important coefficients and most important coefficients similarly. When someone needs to perform the subset selection while applying asymmetric shrinkage on coefficients, the adaptive lasso can be used, and if someone wants to retain all the variables in the model he/she can use generalized ridge regression. However, since the generalized ridge regression is defined with a transformed design matrix, and since it uses a number of tuning parameters which is equal to the number of predictors in the model, the user has no control over the amount of shrinkage on any individual coefficient. This thesis addresses this issue by proposing some new regularized methods which can be used as alternatives to ridge regression, following the idea of the non-negative garrote function. First, we develop the quadratic garrote, which shrinks coefficients unequally, and at the same time retains all the variables in the model. We show that this approach, it is capable of shrinking smaller coefficients even faster than the adaptive lasso while keeping the larger coefficients almost untouched. Also, we generalize the quadratic garrote so that it gives the flexibility for the user to decide the level of shrinkage on each variable directly, based on his experience or prior knowledge. We derive a closed-form solution for the quadratic garrote problem and study the theoretical properties of the suggested estimator such as its variance, expectation, and bias.

Furthermore, we explore the possibility of using different loss functions as the penalty in the non-negative garrote and develop the LINEX regression method as a novel shrinkage approach. We use a numerical optimization technique to estimate the LINEX regression coefficients. In addition, through simulation studies under different settings as well as real world applications, we show that the suggested shrinkage method can be used as a better substitute for the ridge regression in terms of prediction error and practical use.

Keywords: LINEX regression, Multiple linear regression, Non-negative garrote, Shrinkage methods

Acknowledgment

Above all, I would like to express my very profound gratitude to my supervisors, Dr. Saman Muthukumarana and Dr. Mohammad Jafari Jozani for their invaluable guidance, encouragement, and all the support at each step of my thesis and during the master's degree program. I am indebted to them for sharing their knowledge and experience with me, not only to complete the thesis successfully but also towards my future success.

Besides my advisors, I am most grateful to all the members of my thesis committee, Dr. Liqun Wang and Dr. Julien Arino for reading my thesis and for their valuable comments.

A very special gratitude goes out to all of my friends and staff at the department of statistics for their cheerful support throughout my master's degree life.

Finally, I must thank my wife and my parents for their countless love and encouragement.

Dedication

In dedication to my family.

Contents

List of Tables

List of Figures

Chapter 1

Motivation and Thesis Overview

In this chapter, we outline the theory of the multiple linear regression model estimation with ordinary least squares (OLS) approach. We point out that the OLS estimates are not appropriate under certain conditions and then we brief some of the most commonly used remedies such as subset selection, principal component regression and shrinkage methods to overcome the drawbacks of OLS estimates. Furthermore, we emphasize the necessity of novel shrinkage approaches which perform asymmetric shrinkage on regression coefficients.

1.1 Introduction

Multiple regression model is one of the well known statistical tools which is commonly used in many applications. The standard multiple linear regression model can be written as

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \quad i = 1, 2, ..., n, \tag{1.1}$$

where, $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^\top$ is the vector of unknown coefficients, y_i is the response value for the i^{th} observation and $x_{i1}, ..., x_{ip}$ are corresponding values at p explanatory (predictor) variables. We assume ϵ_i's to be independent and identically distributed (iid) with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$. In matrix form we can write the multiple linear regression model as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.2}$$

where, $\boldsymbol{y}$ is the response vector and $\boldsymbol{X}$ is the $n \times (p+1)$ design matrix where the first column consists of $1'$s associated with β_0 in (1.1). Furthermore, $\boldsymbol{\epsilon}$ is the vector of random errors with $E(\boldsymbol{\epsilon}) = \boldsymbol{0}$ and $Var(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}_n$, where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix.

The most widely used method for estimating β_j's is the ordinary least squares technique which minimizes sum of squared errors (SSE) given by

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2, \\ &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \end{aligned} \tag{1.3}$$

where, $\boldsymbol{A}^\top$ denotes the transpose of $\boldsymbol{A}$. Given that $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}$ exists, the solution to the above minimization problem can be obtained as

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

where $\hat{\boldsymbol{\beta}}$ is the vector of least squares estimates and $\boldsymbol{y}$ is the vector of observed response values. Under the assumptions of zero mean and constant variance of residuals, OLS estimates have many good properties which have made them popular among statisticians. For example, OLS models are easy to compute and they are easy to interpret. Furthermore, they are scale invariant. That is, suppose we measure the room temperature in Celsius and use it as a predictor variable. Someone else can measure it in Fahrenheit. This scale change will not affect the least squares coefficient estimates. The coefficient of the first model will be simply a scaled version of the corresponding coefficient of the second model. According to the Gauss-Markov Theorem, OLS estimator $\hat{\beta}_j$ is the Best Linear Unbiased Estimator (BLUE) for the true population parameter β_j (Rao et al., 2008). That is, it is unbiased and, has the least variance among unbiased linear estimators.

1.2 Why not OLS?

As pointed out by Horel (1962), OLE estimates might not be the best choice when multicollinearity is present in data. Multicollinearity refers to the presence of large correlations between the explanatory variables of a model. Multicollinearity issue is frequent in data from social experiments, consumer surveys, market research, medical research, etc. As a consequence of multicollinearity problem, OLS estimates tend to have large standard errors which leads to highly unstable coefficients. Hence, a small change in the dataset will result in a large change in coefficient estimates. Large standard errors will lead to statistically non-significant coefficient estimates, and the corresponding confidence intervals for the true regression coefficients will be wide. Furthermore, OLS estimates can have unexpected signs and magnitudes which can lead to wrong conclusions (Yoo et al., 2014).

To understand how multicollinearity affects the prediction accuracy, consider a design matrix $\boldsymbol{X}$, where the explanatory variables are standardized. If predictors are uncorrelated, $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_p$, where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. Consider the mean squared error (MSE) of OLS estimators given by

$$\begin{aligned} \text{MSE}(\hat{\boldsymbol{\beta}}) &= E\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right], \\ &= \sigma^2 \text{Trace}[(\boldsymbol{X}^\top \boldsymbol{X})^{-1}], \\ &= \sigma^2 \sum_{j=1}^{p} \frac{1}{\lambda_j}, \end{aligned} \tag{1.4}$$

where λ_j is the j^{th} eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$. When $\boldsymbol{X}^\top \boldsymbol{X}$ deviates from $\boldsymbol{I}_p$ and predictors are correlated, some eigenvalues will be very small. Consequently, MSE will take very large values and, OLS estimates are not reliable.

Two of the classical methods to deal with multicollinearity problems are subset selection and principal component regression. Subset selection results in much simpler models with better interpretability by selecting a subset of predictor variables which best explains the response. By selecting a subset, we can avoid correlated coefficient appear in the model simultaneously. However, in high dimensional setting, where we have a large number of predictors, selecting the best subset is computationally challenging. As an example, consider a situation with 50 predictors, which corresponds to $2^{50} = 1.1 \times 10^{15}$ possible models. Hence, selecting the best subset of variables is computationally inefficient, and sometimes is even impossible. Instead of trying to eliminate variables, we can use dimension reduction techniques such as principal component (PC) to reduce the dimension. In this approach, we use a linear transformation of all predictor variables to create a new set of variables called principal components (PCs). These PCs are constructed to be orthogonal to each other, hence their multicollinearity will not be an issue any more. By selecting the first k ($< p$) PCs which describe the largest variability in the predictors, PC regression method will give a remedy for the high dimensional problem. However, PC regression models suffer from interpretability of model parameters. Also, as each principal component is a linear combination of all predictors, it is very difficult to identify the most important variables in predicting the response Y through PC regression model.

1.3 Alternatives to OLS Method

Researchers have been actively developing estimation methods to deal with the multicollinearity issue. One approach is to allow for bias and develop bias estimators that are more stable than OLS estimators. Regularized least squares estimation approach (Shrinkage method) is a class of such methods which

does the coefficient estimation in a regularized manner as a remedy for the inflation and the instability associated with coefficient estimates. As a result of the regularization, parameter estimates are shrunken towards zero. Ridge regression is one of the most popular shrinkage methods, which was first suggested by Horel (1962). The idea is to obtain more precise estimates of model parameters by simply adding a small positive quantity to each diagonal element of $\boldsymbol{X}^\top \boldsymbol{X}$. This will add some bias to coefficient estimates. Ridge regression will be discussed in details in Chapter 2.

Least absolute shrinkage and selection operator (the lasso), which was put forward by Tibshirani (1996) is another shrinkage method specially developed to overcome some of the major issues in linear model fitting in high dimensional setting. High dimensional datasets usually contain thousands to billions of predictors (features) with a comparatively small amount of observations. One of the famous examples of high dimensional data is microarray data. Usually, microarray datasets contain thousands of gene expressions with only a few hundreds of observations. When $n < p$, OLS estimates are not unique. Multicollinearity problem is extreme in the high dimensional setting (James et al., 2013). As we mentioned earlier, the subset selection is also not practical for large p. The lasso is one of the most famous methods which is being widely used in high dimensional regression problems. The most interesting feature of the lasso is its ability to make sparse models by performing both shrinkage and subset selection simultaneously. The concept of the lasso has opened the door for new developments in constructing sparse models using other tools such as the elastic net, etc. Another well-known method which has similar characteristics to lasso is the non-negative garrote (NNG) method of Breiman (1995). The idea of the NNG estimation is to adjust the least squares coefficients by multiplying them with non-negative constants to obtain more accurate coefficient estimates. A thorough description of shrinkage methods can be found in Chapter 2.

1.4 Motivation

The most commonly used regularized regression methods, including ridge regression and lasso have a common weakness. That is, they do not consider the importance of each variable when applying the shrinkage, and all the model coefficients will be shrunken towards zero at a similar rate. As an example, applying ridge regression to estimate a model to predict severity of a heart disease, will result in a similar shrinkage on least important coefficients and most important coefficients. The user has no control over the amount of shrinkage on any coefficient. What if the researcher wants a less shrinkage on the larger coefficients, and instead he wants to apply more shrinkage on the least important ones. Suppose due to theoretical and/or practical justifications, we do not want to shrink a set of variables at

all. For instance, assume the researcher who deals with heart disease data knows that exercise level and stress level of the individual have direct impacts on the severity of the heart disease, and he does not want to shrink the effect of those variables. This cannot be achieved with any of the aforementioned regularized regression methods.

There are several other shrinkage methods which were developed to address some of the above mentioned concerns. As an example, the generalized ridge regression approach suggested in Hoerl and Kennard (1970) can be used to apply unequal shrinkage on different coefficients. However, the way that the generalize ridge approach defines the weights on coefficients does not provide any flexibility for the user to decide which variables should be shrinked more or less. The adaptive lasso suggested by Zou (2006) is a more user friendly method in which the user can define the amount of shrinkage on each coefficient. The adaptive lasso usually penalizes the coefficients inversely proportional to their size. That is, it applies less shrinkage on larger coefficients while applying more shrinkage on smaller coefficients. Because of the subset selection property of the adaptive lasso, more smaller coefficients will be set to exactly zero than the regular lasso approach. However, the adaptive lasso is not appropriate if someone wants to retain all the variables in the model.

Following the non-negative garrote idea, Breiman (1995) suggests an approach (quadratic garrote) which can shrink coefficients unequally, and at the same time retains all the variables in the model. By observing the nature of the quadratic garrote penalty, we see that the quadratic garrote is capable of shrinking smaller coefficients even faster than the adaptive lasso while keeping the larger coefficients almost untouched. However, we do not find any publication in the literature which has further implemented the idea. Hence, as a part of this thesis, we implement the quadratic garrote idea, and generalize the quadratic garrote so that it gives the flexibility for the user to decide the level of shrinkage on each variable directly, based on his experience or prior knowledge.

We can observe that the ridge regression, the lasso and the elastic net penalties are symmetric in nature. This is why they apply similar shrinkage on each coefficient. On the other hand, the methods like the generalize ridge regression and adaptive lasso use asymmetric penalties. Because of this, they can treat each coefficient differently. In the recent literature, we find some work on using asymmetric loss functions in the penalty term. For instant, the group exponential lasso which was suggested by Breheny (2015), uses the exponential penalty on the grouped lasso to give exponentially decaying weights to the coefficients within the groups. In this thesis, we further study the possibility of using various loss functions in place of the penalty term in the non-negative garrote. To this end, we propose the LINEX regression as a novel shrinkage approach, where we consider the asymmetric LINEX loss function instead of the penalty term of the non-negative garrote and study the properties of estimated models under

different settings.

1.5 Organization of the Thesis

The organization of this thesis is as follows. In chapter Chapter 2, we provide an overview of some of the popular shrinkage methods including the related literature in details. In particular, we discuss subset selection, ridge regression, generalized ridge regression, the lasso, the elastic net, adaptive lasso and the non-negative garrote methods. Furthermore, we present an example to illustrate aforementioned shrinkage methods.

In Chapter 3, we study the quadratic garrote method suggested by Breiman (1995) and further generalize it to obtain a more practically sound shrinkage approach where the user have some control over the amount of shrinkage on each regression coefficient estimate. There, we study the properties of the suggested estimators theoretically and using simulation studies. Also, we study the performance of the suggested shrinkage methods with a real data example.

In Chapter 4, we consider the linear regression method and study another interesting shrinkage method using the asymmetric linex loss function as the penalty in conjunction with SSE through a constrained optimization problem that needs to be solved numerically. Furthermore, we study the properties of the suggested shrinkage approach using simulation studies and a real data example.

Finally, in Chapter 5 we provide concluding remarks followed by conclusions and a discussion. Also we outline some future related directions.

Chapter 2

Shrinkage Methods

In this chapter, we present the literature of some of the most widely used shrinkage methods which are related to our thesis. Furthermore, we discuss model selection methods and prediction error estimation followed by an example.

2.1 Introduction

As we mentioned in Chapter 1, ridge regression, the lasso and non-negative garrote are the most popular shrinkage methods developed in the recent literature of linear regression. The main purpose of all shrinkage methods is to get out of the class of unbiased estimators by allowing for some bias in order to obtain estimators with less variability and improve the predictability of the model. This is often done by shrinking the parameter estimates towards zero through adding some penalties to the sum of squared errors. This is similar to OLS method, but instead of minimizing the SSE, we minimize SSE under some constraints. Let $\tilde{\boldsymbol{\beta}}$ be the vector of estimated coefficients with a shrinkage method. Then, $\tilde{\boldsymbol{\beta}}$ is obtained as

$$\tilde{\boldsymbol{\beta}} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \right\} \text{ subject to } \sum_{j=1}^{p} f(\beta_j) \leq s, \tag{2.1}$$

where, $f(\beta_j) \geq 0$ is a function of β_j. Usually, we write (2.1) using the Lagrange multiplier approach as

$$\tilde{\boldsymbol{\beta}} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} f(\beta_j) \right\}. \tag{2.2}$$

Here, $\sum_{j=1}^{p} f(\beta_j)$ is called the penalty and λ is called the tuning parameter, which defines the amount of shrinkage. Note that, λ in (2.2) has an inverse relationship with s in (2.1). As we increase λ

(equivalently; reducing s), we obtain much smaller estimates as those under OLS. In other words, as $\lambda \to \infty$, $\tilde{\boldsymbol{\beta}} \to 0$, and as $\lambda \to 0$, $\tilde{\boldsymbol{\beta}} \to \hat{\boldsymbol{\beta}}$ where $\hat{\boldsymbol{\beta}}$ is the usual OLS estimate of $\boldsymbol{\beta}$. That is, for a very large value of λ, all the coefficient estimates will be almost zero and for a very small λ, the coefficient estimates will be approximately equal to OLS estimates. By changing $f(\beta_j)$, we can change the nature of the shrinkage. In the preceding sections, you will observe how the change of $f(\beta_j)$ affects the coefficient estimates.

2.2 Ridge Regression

As we discussed in the introduction, ridge regression is one of the classical methods to deal with multicollinearity and issues arising with high dimensional data. The first motivation for ridge regression is found in Horel (1962) as well as Hoerl and Kennard (1970), where they suggested to use $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p); \lambda \geq 0$ instead of $\boldsymbol{X}^\top \boldsymbol{X}$ in the least squares estimation. Their idea was to control the variance inflation and general instability in the least squares estimates by imposing a small bias. Ridge regression problem can be written as a constrained estimation problem where the idea is to obtain $\hat{\boldsymbol{\beta}}^R$ as estimates of $\boldsymbol{\beta}$ through the following minimization problem:

$$\hat{\boldsymbol{\beta}}^R = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 \right\} \text{ subject to } \sum_{j=1}^{p} {\beta_j}^2 \leq s. \tag{2.3}$$

That is, simply adding a penalty on the size of regression coefficients. $\sum_{j=1}^{p} {\beta_j}^2$ is the squared distance of the coefficient vector $\boldsymbol{\beta}$ from the origin, which is also referred to as the L_2 norm denoted by $||\boldsymbol{\beta}||_2^2$. For the two predictor case with $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$, the nature of the constraint and the role of s can be visually illustrated as in Figure 2.1. As we reduce s, the radius of the circle decreases and the ridge solutions are forced to have smaller values which are on the margin of the circle. As we increase s, the coefficient estimates are less restricted and for a considerably large s, the ridge estimates can be the same as OLS estimates.

Minimization problem of the ridge regression can be also written as

$$\hat{\boldsymbol{\beta}}^R = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2 \right\}. \tag{2.4}$$

Here, λ is called the tuning parameter which defines the amount of shrinkage on the estimated coefficients. As we increase λ, the size of the penalty increases and coefficients are shrunk towards zero. Usually, we

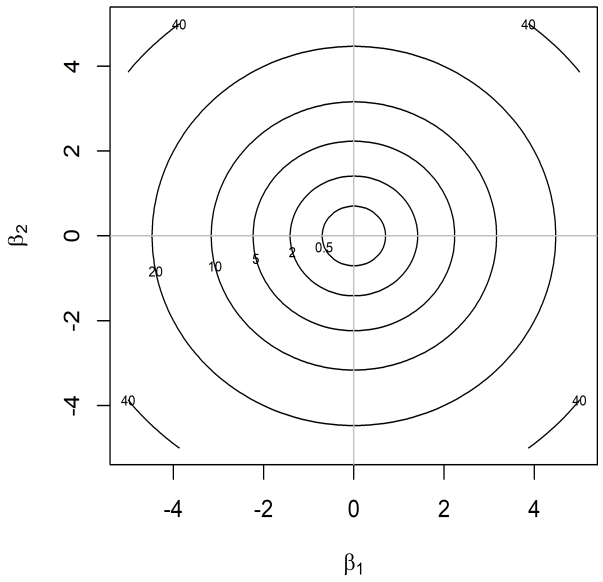


Figure 2.1: Ridge constraints for different s values. Each circle represents those $(\beta_1, \beta_2) \in \mathbb{R}$ such that $\beta_1^2 + \beta_2^2 = s, s \in \{0.5, 2, 5, 10, 20\}$.

do not shrink the intercept β_0. Without loss of generality, we will assume that $\beta_0 = 0$, or $\boldsymbol{X}$ and $\boldsymbol{y}$ are centered. Thus the dimensions of $\boldsymbol{X}$ is $n \times p$. Then the ridge regression problem can be written in matrix form as below

$$\hat{\boldsymbol{\beta}}^R = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta} \right\}. \tag{2.5}$$

The solution to this minimization problem can be obtained as

$$\hat{\boldsymbol{\beta}}^R = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \tag{2.6}$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. To see this, let

$$Q(\boldsymbol{\beta}, \lambda) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}. \tag{2.7}$$

Consider the first derivative of (2.7) with respect to $\boldsymbol{\beta}$

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta}.$$

Since $Q(\boldsymbol{\beta}, \lambda)$ is convex, there exist a unique minimum such that

$$\left. \frac{\partial Q}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}^R} = 0. \tag{2.8}$$

By solving (2.8), we obtain

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)\hat{\boldsymbol{\beta}}^R = \boldsymbol{X}^\top \boldsymbol{y}.$$

By further simplifying, we can obtain a closed form solution for $\hat{\boldsymbol{\beta}}^R$ as

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

Finally, it turns out that the ridge estimation process is nothing but simply adding a constant term to each of the diagonal elements of $\boldsymbol{X}^\top \boldsymbol{X}$. This will remove any singularity issue with $\boldsymbol{X}^\top \boldsymbol{X}$ and, will result in biased estimates for $\boldsymbol{\beta}$. However, the variance associated with the estimates will dramatically decrease by adding a little bias. Thus, it is important to consider the bias-variance trade-off when obtaining the ridge estimates. Selection of the tuning parameter λ is usually done by selecting a sequence of λ values and evaluating the prediction error at each value of λ with cross-validation method which we will discuss in Section 2.6.

To have further insight, we brief some theory in Hoerl and Kennard (1970), on ridge regression. Let $\tilde{\boldsymbol{\beta}}$ be any estimator of the vector of true population coefficients $\boldsymbol{\beta}$ and let $\hat{\boldsymbol{\beta}}$ be the vector of least squares estimators. Then, SSE of $\tilde{\boldsymbol{\beta}}$ can be written as

$$\begin{aligned}
\text{SSE}(\tilde{\boldsymbol{\beta}}) &= (\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^\top (\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}), \\
&= \left[(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) + (\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})\right]^\top \left[(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) + (\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})\right], \\
&= (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^\top (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{X}^\top \boldsymbol{X}(\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}) - 2(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^\top (\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}).
\end{aligned}$$

We can easily show that $(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^\top (\boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}) = 0$. Hence,

$$\begin{aligned}
\text{SSE}(\tilde{\boldsymbol{\beta}}) &= (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^\top (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) + (\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{X}^\top \boldsymbol{X}(\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}), \\
&= \text{SSE}(\hat{\boldsymbol{\beta}}) + \varphi(\tilde{\boldsymbol{\beta}}).
\end{aligned} \tag{2.9}$$

$\text{SSE}(\tilde{\boldsymbol{\beta}})$ will take its minimum value when $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$. $\text{SSE}(\hat{\boldsymbol{\beta}})$ is constant and hence, the contours of $\text{SSE}(\tilde{\boldsymbol{\beta}})$ will be hyperellipsoids centered at $\hat{\boldsymbol{\beta}}$. Let $\varphi_0 > 0$ to be a fixed value for $\varphi(\tilde{\boldsymbol{\beta}})$. Then for $\varphi_0 > 0$, there will be a continuum of possible values for $\tilde{\boldsymbol{\beta}}$ which will result the same SSE. From the $\tilde{\boldsymbol{\beta}}$'s perspective, as $\tilde{\boldsymbol{\beta}}$ deviates from OLS estimates, $\text{SSE}(\tilde{\boldsymbol{\beta}})$ increases.

As we mentioned earlier, in ridge regression we restrict the parameter space by imposing a penalty on the squared length of the parameter estimates and then selecting the best estimate in the restricted space.

To understand the idea, let's consider an example with two predictors x_1 and x_2, which has been plotted in Figure 2.2. In Figure 2.2, the observations are generated from a multivariate normal distribution with mean vector $\begin{pmatrix} 5 \\ 10 \end{pmatrix}$ and the covariance matrix $\begin{pmatrix} 2 & 1.8 \\ 1.8 & 3 \end{pmatrix}$. The response y was generated from the model $Y = 10Z_1 + 20Z_2 + \epsilon$, with $\epsilon \sim N(0, 15)$ where Z_1, Z_2 are the standardized predictors. The contour plot of SSE($\boldsymbol{\beta}$) is given in Figure 2.3 along with the OLS estimates and the ridge solution for a fixed s. The ridge estimate on the plot was evaluated with the best s using 10-fold cross-validation errors. As we observe, the ridge estimates are much smaller than the least squares estimates and they lie on the boundary of the ridge penalty function shown by the circle. That is, for a given s (or λ), the ridge estimates can be found at the point where the first contours of SSE($\boldsymbol{\beta}$) (ellipse) touches the boundary of the penalty region.

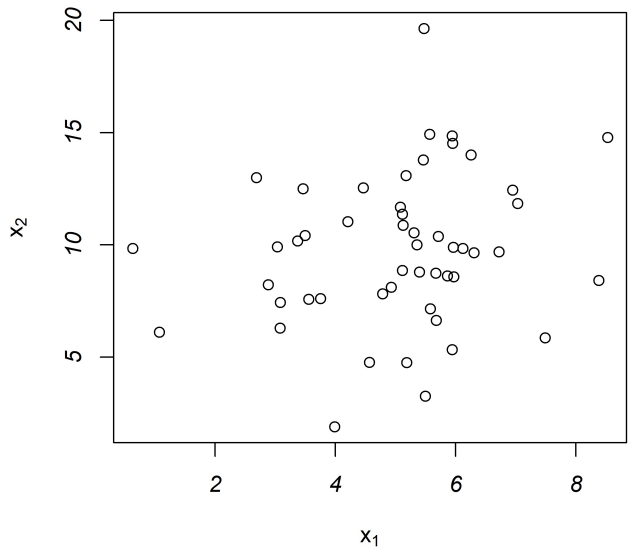


Figure 2.2: Scatter plot of a example of size $n = 50$ on two predictors x_1 and x_2.

To have more insight into the nature of the shrinkage done by ridge regression, suppose the design matrix is orthonormal, that is $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_p$. Then the ridge solution can be written as

$$\hat{\boldsymbol{\beta}}^R = \frac{1}{(1+\lambda)} \boldsymbol{X}^\top \boldsymbol{y}.$$

For the orthonormal case, OLS estimates are given by $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\top \boldsymbol{y}$ and hence,

$$\hat{\beta}_j^R = \frac{1}{(1+\lambda)} \hat{\beta}_j.$$

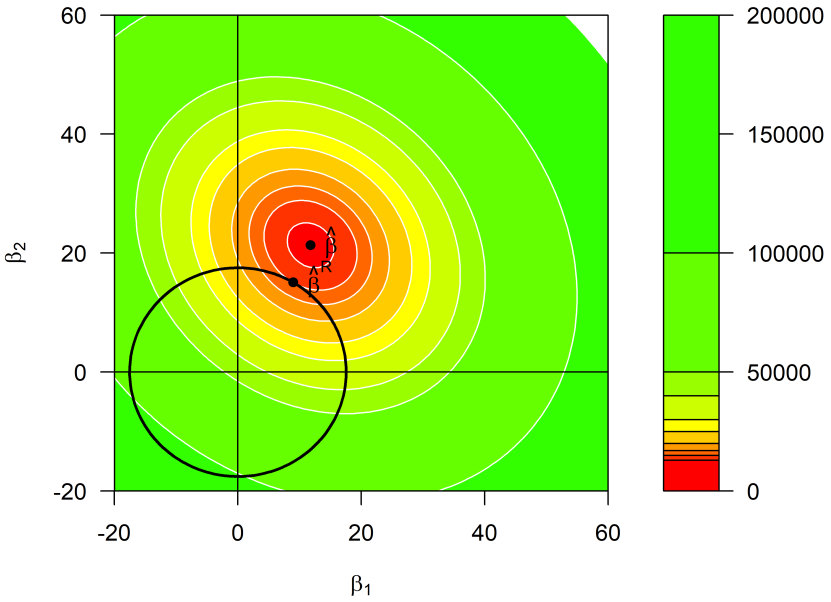


Figure 2.3: Contour plot of SSE with the ridge penalty and the solution. Here, $\hat{\boldsymbol{\beta}}$ denotes the OLS estimate of $\boldsymbol{\beta}$ while $\hat{\boldsymbol{\beta}}^{\boldsymbol{R}}$ is the corresponding ridge estimate for specific value of λ.

Here, $1/(1+\lambda)$ is called the shrinkage factor and it is a constant. Hence, regardless of the size of the coefficient estimate, ridge regression applies a constant level of shrinkage to all coefficients. Further theoretical properties and proofs related to ridge regression are followed by the theory of general quadratic garrote, which is a generalized version of ridge regression, presented in Chapter 3. Hence, we will not present them here. However, one thing to keep in mind is that, since ridge regression does the shrinkage by imposing a penalty on the size of each coefficient, any change of the scale of the predictors will significantly affect the results. Hence, it is advisable to standardize the predictors prior to applying ridge regression methodology. This concern is the same for the lasso and the elastic net as well.

2.2.1 Generalized Ridge Regression

In ridge regression, the single tuning parameter λ defines the amount of shrinkage on the estimated coefficients. Hoerl and Kennard (1970) also suggested a general form of ridge regression which uses p number of tuning parameters instead of a single λ. In the previous section, we saw that the ridge regression applies a constant level of shrinkage to each coefficient. By using p tuning parameters instead of a single tuning parameter, one can introduce different levels of shrinkage on each coefficient. Hoerl and Kennard (1970) defines the generalized ridge regression in a orthogonal basis created by the eigenvector space of $\boldsymbol{X}^\top \boldsymbol{X}$. Let $\boldsymbol{P}$ be the orthogonal matrix where the columns of $\boldsymbol{P}$ are the

eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ and $\boldsymbol{\Lambda}$ be a diagonal matrix whose diagonal elements are the eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$. Then, $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^\top$. Define $\boldsymbol{X}^* = \boldsymbol{X}\boldsymbol{P}$, and $\boldsymbol{\alpha} = \boldsymbol{P}\boldsymbol{\beta}$. Then,

$$\boldsymbol{Y} = \boldsymbol{X}^*\boldsymbol{\alpha} + \boldsymbol{\epsilon}.$$

Then, the ridge estimator is given by

$$\begin{aligned}\hat{\boldsymbol{\alpha}}^R &= \left[\boldsymbol{X}^{*\top}\boldsymbol{X}^* + \lambda\boldsymbol{I}_p\right]^{-1}\boldsymbol{X}^{*\top}\boldsymbol{y},\\ &= \left[\boldsymbol{P}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{P} + \lambda\boldsymbol{I}_p\right]^{-1}\boldsymbol{X}^{*\top}\boldsymbol{y},\\ &= \left[\boldsymbol{\Lambda} + \lambda\boldsymbol{I}_p\right]^{-1}\boldsymbol{X}^{*\top}\boldsymbol{y}. \qquad (2.10)\end{aligned}$$

Since $\boldsymbol{b} = \boldsymbol{X}^{*\top}\boldsymbol{y}$. Then

$$\alpha_i^R = \frac{1}{\delta_i + \lambda}b_i, \qquad (2.11)$$

where b_i is the i^{th} entry of $\boldsymbol{b}$ and, δ_i is the i^{th} eigenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$ (or i^{th} diagonal element of $\boldsymbol{\Lambda}$). Ridge regression adds the same λ to each eigenvalue in the denominator to estimate α_i. Boer and Hafner (2005) points outs that, in order to obtain more stable coefficients, simply adding a constant to each eigenvalue is not suitable. Hence, it is more reasonable to add different constants in place of λ to obtain

$$\alpha_i^R = \frac{1}{\delta_i + k_i}b_i. \qquad (2.12)$$

Following the above idea, the generalized ridge estimator is defined as

$$\hat{\boldsymbol{\alpha}}^{GR} = \left[\boldsymbol{X}^{*\top}\boldsymbol{X}^* + \boldsymbol{K}\right]^{-1}\boldsymbol{X}^{*\top}\boldsymbol{y}, \qquad (2.13)$$

where $\boldsymbol{K}$ is a $p \times p$ diagonal matrix consists of diagonal elements $k_1, k_2, ..., k_p$. Above estimator can be considered as the solution for the optimization problem

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{(\boldsymbol{y} - \boldsymbol{X}^*\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}^*\boldsymbol{\beta}) + \boldsymbol{\beta}^\top\boldsymbol{K}\boldsymbol{\beta}\right\}. \qquad (2.14)$$

The MSE of $\hat{\boldsymbol{\alpha}}^{GR}$ is given by

$$MSE(\hat{\boldsymbol{\alpha}}^{GR}) = (\hat{\boldsymbol{\alpha}}^{GR} - \boldsymbol{\alpha})^\top(\hat{\boldsymbol{\alpha}}^{GR} - \boldsymbol{\alpha}). \qquad (2.15)$$

With some simplifications, it can be shown that

$$MSE(\hat{\boldsymbol{\alpha}}^{GR}) = \sum_{i=1}^{p} \frac{\sigma^2\delta_i + \alpha_i^2 k_i^2}{(\delta_i + k_i)^2}. \tag{2.16}$$

By taking the derivative of (2.16) with respect to k_i's we have

$$\frac{\partial}{\partial k_i} MSE(\hat{\boldsymbol{\alpha}}^{GR}) = \frac{2\delta_i(\alpha_i^2 k_i - \sigma^2)}{(\delta_i + k_i)^3}. \tag{2.17}$$

At the optimum k_i, $\frac{\partial}{\partial k_i} MSE(\hat{\boldsymbol{\alpha}}^{GR}) = 0$. Since $\boldsymbol{X}^\top \boldsymbol{X}$ is full ranked, $\delta_i > 0$ for all $i = 1, 2, \ldots, p$. Taking k_i to be non-negative, we can derive

$$k_i = \frac{\sigma^2}{\alpha_i^2} \tag{2.18}$$

to be the optimal solution for k_i (Hemmerle, 1975). However, this is not feasible since α_i is unknown (Boer and Hafner, 2005). Hence, Hoerl and Kennard (1970) suggested an iterative approach to estimate k_i, which is initiated at $\hat{\sigma}^2/\hat{\alpha}_i^2$, where $\hat{\sigma}^2$ is the estimated error variance and $\hat{\alpha}_i^2$ is the OLS estimate of α_i.

2.3 The Lasso

Least absolute shrinkage and selection operator (the lasso), which was first proposed by Tibshirani (1996), can be identified as the most popular shrinkage method today. The lasso coefficients, $\hat{\boldsymbol{\beta}}^L$ are obtained as

$$\hat{\boldsymbol{\beta}}^L = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \right\} \text{ suject to } \sum_{j=1}^{p} |\beta_j| \leq s. \tag{2.19}$$

Using the Lagrange multiplier approach, we can re-write the above problem as

$$\hat{\boldsymbol{\beta}}^L = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{2.20}$$

Here, λ is the tuning parameter which defines the amount of shrinkage on the coefficient estimates and is usually estimated with 5 or 10-fold cross-validation approach. Similar to the ridge regression problem, lasso does model fitting under a constraint based on the size of the parameter estimates. However, instead of the L_2 constraint, lasso imposes the L_1 constraint (first norm of the coefficient vector) on the estimates. The lasso does the shrinkage since it still has a penalty based on the size of the coefficients. By introducing the L_1 norm instead of L_2 norm, lasso gains an additional attractive property of subset

selection. While shrinking parameters towards zero, lasso sets some of the coefficients exactly to zero; thus producing sparse models by performing variable selection. This property is extremely useful in high dimensional settings. As we mentioned in Chapter 1, subset selection in the high dimensional setting is not practical, and for $n < p$ case, no unique OLS estimates exist. But, lasso can still be used to obtain coefficient estimates and most importantly, it will produce a much simpler interpretable model with a small number of predictors considered to be the most important variables in predicting the response.

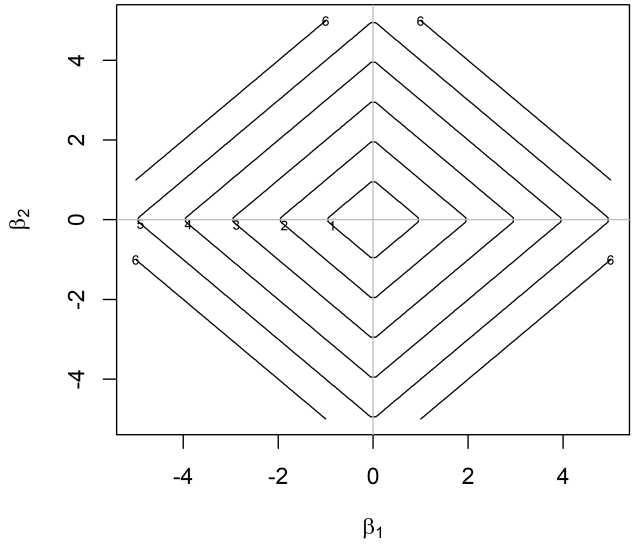


Figure 2.4: The lasso constraint $|\beta_1| + |\beta_2| \leq s$ for different s values.

To understand how lasso gains subset selection ability, we have to study the nature of the constraint formed by the lasso penalty. Let's again consider multiple linear regression model with two predictors as discussed in Section 2.1. The lasso penalty for different s values are shown in Figure 2.4. As we increase s, area of the constrained region increases. The lasso solution along with the penalty and contours of SSE has been presented in Figure 2.5 for the same dataset that we used in Section 2.1. The lasso penalty results in a diamond-shaped constrained region and unlike in the ridge, lasso penalty has sharp corners. Hence, if the first place that a contour touches the diamond is at a corner, the method will have one coefficient set exactly to zero. Figure 2.6 shows such a situation where lasso has produced a sparse model. In this case, $\hat{\beta}_1^L$ is exactly zero. In higher dimensions, contours of the SSE hyperellipsoids hitting a corner of the polytope formed by the lasso penalty will result sparse models by setting one or more coefficient estimates to exactly zero.

When the design matrix is orthonormal, one can easily show that the lasso coefficient estimates are

$$\hat{\beta}_j^L = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \frac{\lambda}{2})^+, j = 1, ..., p. \tag{2.21}$$

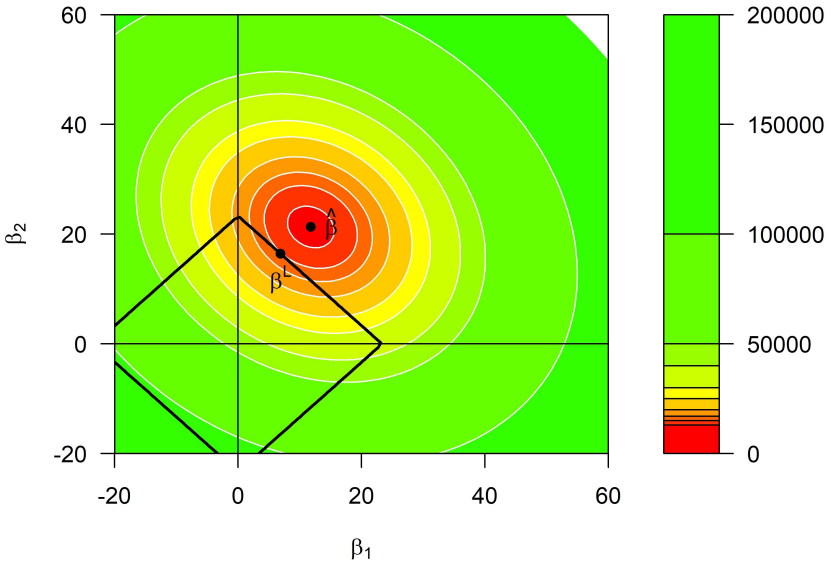


Figure 2.5: Contour plot of SSE with the lasso penalty and the solution.

To see this in vector notations, we can write (2.20) as

$$\begin{aligned} Q(\boldsymbol{\beta}, \lambda) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda |\boldsymbol{\beta}|, \\ &= \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda |\boldsymbol{\beta}|. \end{aligned}$$

Taking the first derivative of Q with respect to $\boldsymbol{\beta}$ leads to

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda \boldsymbol{S},$$

where $\boldsymbol{S}$ is the vector of soft thresholding operators $s_j, j = 1, 2, ..., p$, which is defined as

$$s_j = \begin{cases} \text{sign}(\beta_j) & , \beta_j \neq 0, \\ \text{any value in } [-1, 1] & , \beta_j = 0. \end{cases}$$

Since $Q(\boldsymbol{\beta}, \lambda)$ is convex, there exist a unique minimum at $\hat{\boldsymbol{\beta}}^L$, such that

$$\left. \frac{\partial Q}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}^L} = 0. \tag{2.22}$$

By solving 2.22, we obtain

$$\begin{aligned} \boldsymbol{X}^\top \boldsymbol{X} \hat{\boldsymbol{\beta}}^L &= \boldsymbol{X}^\top \boldsymbol{y} - \frac{\lambda}{2} \boldsymbol{S}, \\ \hat{\boldsymbol{\beta}}^L &= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} (\boldsymbol{X}^\top \boldsymbol{y} - \frac{\lambda}{2} \boldsymbol{S}). \end{aligned}$$

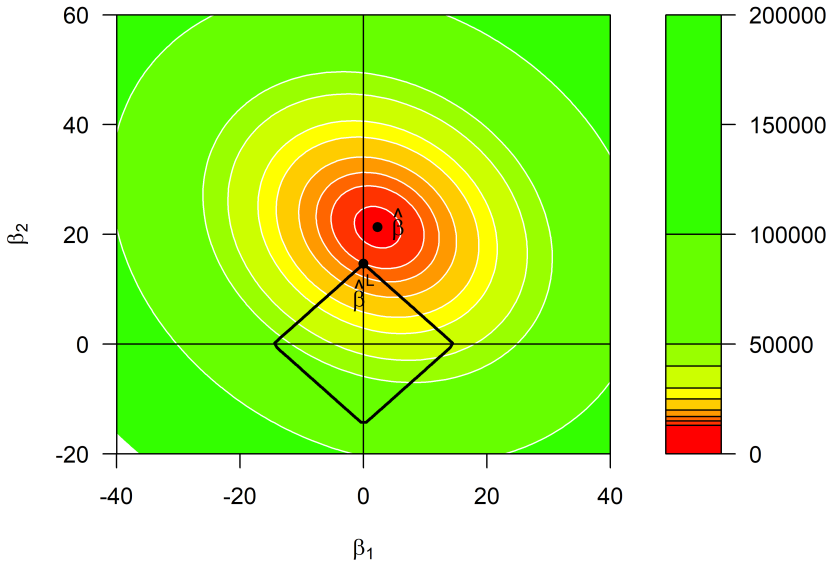


Figure 2.6: Example of a sparse solution given by lasso.

Under the orthonormal assumption, $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_p$ and $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\top \boldsymbol{y}$. Then

$$\hat{\boldsymbol{\beta}}^L = \hat{\boldsymbol{\beta}} - \frac{\lambda}{2}\boldsymbol{S}.$$

Let $\hat{\beta}_j{}^L$ be the j^{th} lasso coefficient estimate. Then,

$$\begin{aligned} \hat{\beta}_j{}^L &= \hat{\beta}_j - \frac{\lambda}{2} s_j, \\ &= \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \frac{\lambda}{2})^+, \end{aligned} \tag{2.23}$$

where $(a)^+ = \max\{0, a\}$.

As we see in Figure 2.7, unlike the ridge regression, lasso translates all OLE estimates by a constant value and truncates at zero. Hence, for all model coefficients where $|\hat{\beta}_j| < \lambda/2$, lasso estimates will be exactly zero. For $p \leq 2$ case, it can be shown that the lasso solutions would have the same signs as OLS estimates. However, for $p > 2$, lasso solutions can have signs different from the OLS estimates (Tibshirani, 1996). For the non-orthonormal design case, we do not have a closed form solution for the lasso optimization problem. Hence, we have to use some numerical method to obtain coefficient estimates. In the recent literature, we can find many efficient algorithms for estimating regression coefficients with the lasso penalty. The least angle regression (LAR) algorithm is one such method suggested by Efron et al. (2004). Coordinate descent algorithm suggested by Wu and Lange (2008) is

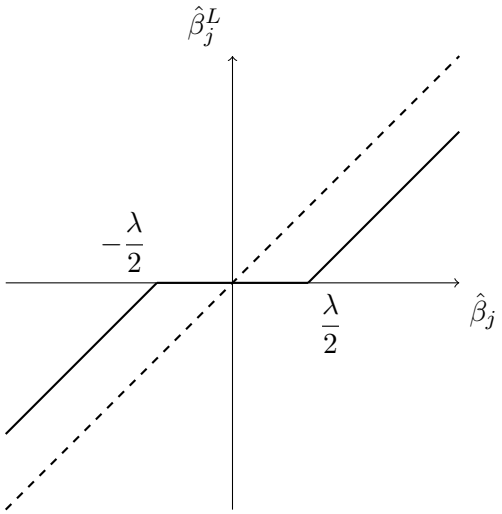


Figure 2.7: Plot of $\hat{\beta}_j^L$ against $\hat{\beta}_j$ under orthonormal design matrix.

another widely used method. Coordinate descent algorithm is identified to be exceptionally fast in estimating regression coefficients with the lasso penalty.

Same as in ridge regression, best λ is usually selected by a cross-validation method. Closed form estimation of standard errors for lasso estimates is not feasible due to the non-linear and non-differentiable nature of the optimizing function. Hence, the common approach is to use the bootstrap method or some approximations.

In the recent literature, many extensions for the lasso can be found. Few of the most popular extensions are the adaptive lasso of Zou (2006), the group lasso of Yuan and Lin (2006), the fused lasso of Tibshirani et al. (2005), the sparse grouped lasso of Simon et al. (2013), etc. The group lasso can be used when we want to select grouped variables together. In this approach, an entire group of variables will be selected based on the strength of all variables in the group and not on the individual strength of each variable. However, the group lasso does not posses any sparsity within groups. Hence, one can use the sparse grouped lasso to have the sparse effect in the group level and within group level. The fused lasso incorporates the ordering of the features when performing subset selection. This method can be useful when the features have a natural order.

2.3.1 The Adaptive Lasso

As we observed in (2.20), the lasso equally penalizes all the coefficients in the L_1 constraint. As shown in Zou (2006), the underling model has to satisfy a non-trivial condition so that the lasso is consistent

for variable selection. Furthermore, the lasso equally penalizes all the coefficients and sometimes this is not fair (Zou, 2006). Thus, as a remedy Zou (2006) proposed to assign different weights to penalize different coefficients in the lasso penalty. The adaptive lasso estimate $\hat{\boldsymbol{\beta}}^{AL}$ is obtained as

$$\hat{\boldsymbol{\beta}}^{AL} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}, \tag{2.24}$$

where $\boldsymbol{w} = \{w_1, w_2 \ldots, w_p\}^\top$ is the weight vector. The penalty (2.24) can be asymmetric when $w_i \neq w_j$ for all $i \neq j$. Figure 2.8 presents the nature of the constraint and solutions for different selection of weights, w_j's. As you can see, the shape of the penalty region changes with the choice of w_j's and thus, the solution changes. The coefficients estimates which got larger weights in the adaptive lasso penalty shrinks more than the coefficients which got smaller weights.

Furthermore, let $\hat{\boldsymbol{\beta}}^*$ be a root-n-consistent estimator to $\boldsymbol{\beta}$ (the OLS estimate $\hat{\boldsymbol{\beta}}$ is consistent for $\boldsymbol{\beta}$). Let $\hat{\boldsymbol{w}} = 1/|\hat{\boldsymbol{\beta}}^*|^\gamma$, where $\gamma > 0$. Then the adaptive lasso is defined as

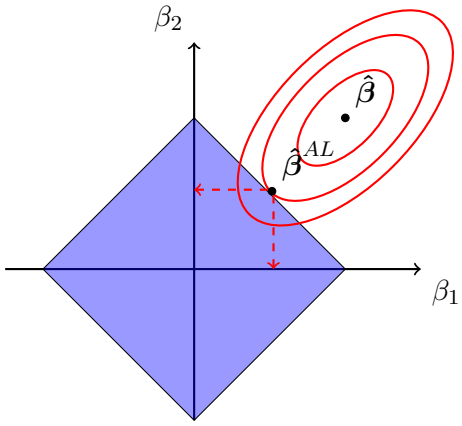
$$\hat{\boldsymbol{\beta}}^{AL} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j| \right\}. \tag{2.25}$$

Zou (2006) suggested the least angle regression (LAR) algorithm to numerically obtain the adaptive lasso coefficient estimates. The adaptive lasso approach consists of two tuning parameters which can be determined with two-dimensional cross-validation. Zou (2006) suggested to use OLS estimates for $\hat{\boldsymbol{\beta}}^*$ unless multicollinearity is present. If multicollinearity is a problem, the ridge estimator can be used for $\hat{\boldsymbol{\beta}}^*$. Otherwise $\hat{\boldsymbol{\beta}}^*$ can be considered as another tuning parameter using an other consistent estimator for $\hat{\boldsymbol{\beta}}^*$ and perform three-dimensional cross-validation to select optimal $(\hat{\boldsymbol{\beta}}^*, \gamma, \lambda)$.

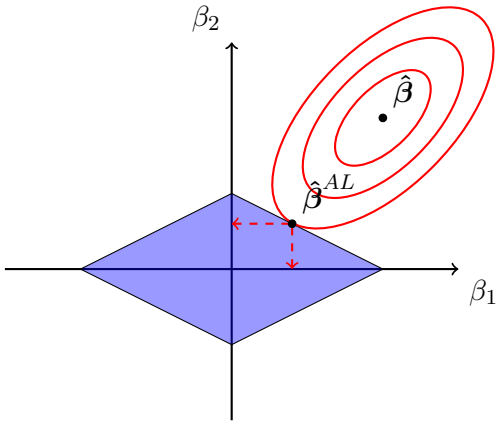
2.4 The Elastic Net

The elastic net is a hybrid approach of ridge regression and the lasso. This was suggested by Zou and Hastie (2005) and it is more flexible than the ridge and/or the lasso. Elastic net is suggested by the authors to overcome three main drawbacks of the lasso.

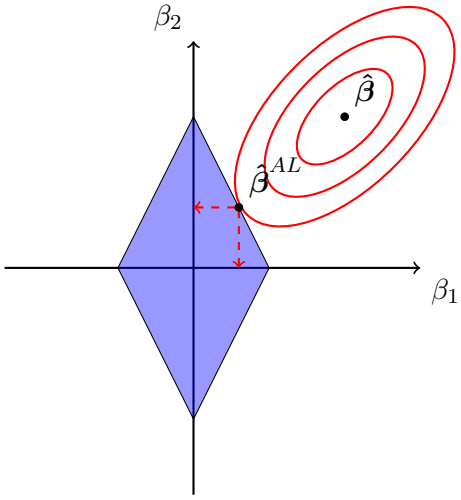
1. The lasso does not perform a satisfactory subset selection when the number of predictors is much larger than sample size ($p >> n$). In this setting, the lasso can select maximum of n variables and then the model saturates.



(a) $|\beta_1| + |\beta_2| \leq s$



(b) $|\beta_1| + 2|\beta_2| \leq s$



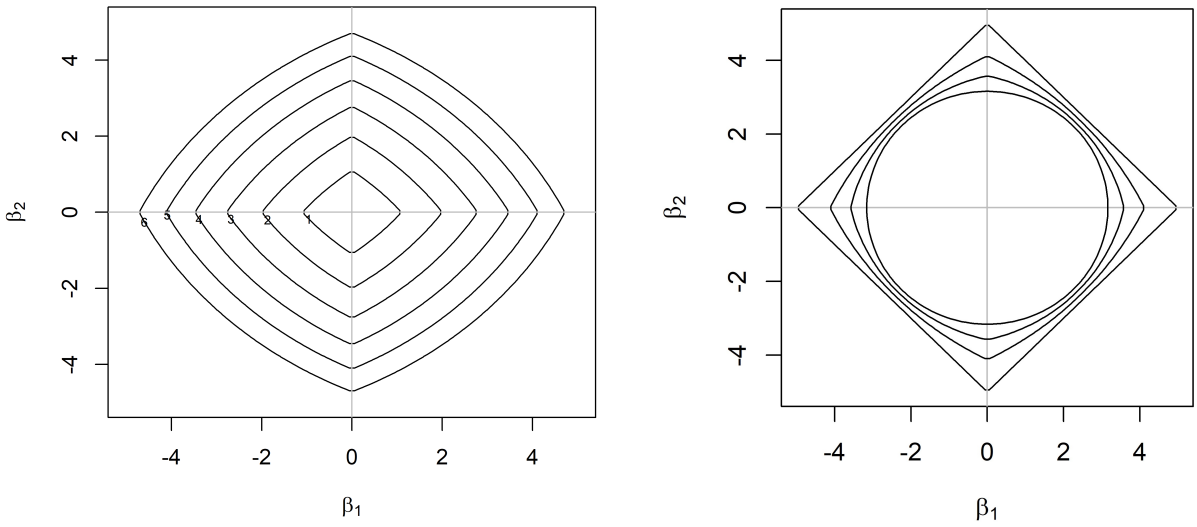
(c) $2|\beta_1| + |\beta_2| \leq s$

Figure 2.8: the adaptive lasso penalty for different choice of w_j's.

2. The lasso does not posses the grouping effect which is usually preferred in practice. That is, when there is a set of variables with very high pairwise correlations, the lasso is likely to select only one variable and omit the rest of the variables in the group.

3. Under the general setting of $n > p$, the lasso is usually outperformed by ridge regression in terms of the prediction accuracy when there is a strong multicollinearity within predictors.

As an example, gene expression data comes with thousands of features (predictors) with only a few (usually less than 100) observations. As we explained, the lasso will not select a sufficient number of genes and also will not do grouped variable selection that is essential in revealing grouping information in genes. Zou and Hastie (2005) showed that the elastic net produces sparse models with better prediction



(a) (b)

Figure 2.9: (a) Elastic net constraint for different s values ($\alpha = 0.5$) ,(b) Elastic net constraint for different α values ($\alpha = 0$ (ridge), $0.5, 0.8, 1$ (lasso)).

accuracy than the lasso while supporting the grouping effect. The elastic net coefficient estimates $\hat{\boldsymbol{\beta}}^{EN}$, are obtained as

$$\hat{\boldsymbol{\beta}}^{EN} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \left[\alpha {\beta_j}^2 + (1-\alpha)|\beta_j| \right] \right\}.$$

Here $0 \leq \alpha \leq 1$ is a constant defined by the user. For $p = 2$, the nature of the penalty can be observed in Figure 2.9a for $\alpha = 0.8$. The constrained region is neither a circle nor a diamond. It has the properties of both the ridge and the lasso penalties with sharp edges. Thus, the elastic net also does subset selection. Figure 2.9b shows the role of α on the penalty for a multiple regression model with two predictors when $s = 5$. As α increases, estimates are much closer to the ridge ones and for small α, the properties are more like those under the lasso. Figure 2.10 presents the solution of the elastic net with $\alpha = 0.8$, for the same dataset with two predictors represented in Figure 2.2.

2.5 Non-negative Garrote (NNG)

The concept of non-negative garrote which was originally suggested in Breiman (1995) as a better subset selection, is really interesting. Tibshirani (1996) mentions that the motivation for the lasso was the non-negative garrote. The shrinkage methods that we discussed so far did the estimation which does

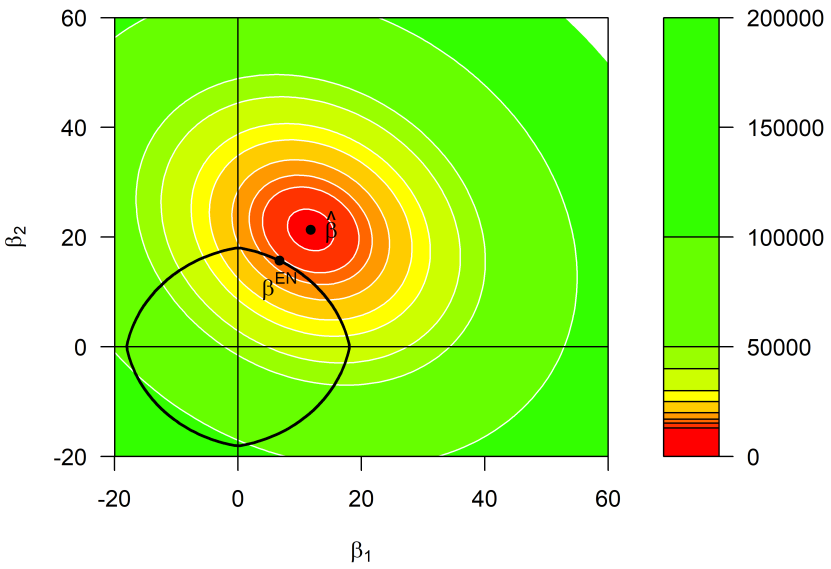


Figure 2.10: Contour plot of SSE and the elastic net penalty.

not involve OLS estimates. It is true that there are certain situations that we cannot rely on OLS estimates or we cannot obtain them at all. However, as we mentioned earlier, OLS estimates posses many statistically desirable characteristics. Hence, it makes sense if someone does not want to ignore OLS estimates when estimating the model parameters. Instead of completely avoiding OLS estimates, we can use non-negative garrote estimation method to adjust OLS estimates in order to achieve a higher prediction accuracy. This is done by adjusting each OLS estimates $\hat{\beta}_j$. To this end, Breiman (1995) proposed to find optimum positive constants $c_j's$ such that $\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} c_j x_{ij} \hat{\beta}_j$ is a better model than OLS. In order to avoid drastic changes to the OLS model he imposed an intuitively sound constrained $\sum_{j=1}^{p} c_j \leq p$. In other words, he formulated his problem as minimizing

$$\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij})^2, \text{ subject to } \sum_{j=1}^{p} c_j \leq s. \tag{2.26}$$

The problem can be re-written with Lagrangian multiplier as the problem of finding

$$(\hat{c}_0, \hat{c}_1, ..., \hat{c}_p) = \underset{c_0, c_1, ..., c_p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij})^2 + \lambda \sum_{j=1}^{p} c_j.$$

Again, $\lambda > 0$ is the tuning parameter that we decide with some prediction error criteria, which is evaluated at a sequence of values for λ and then selecting λ which gives the least prediction error. In Section 2.6, we present some of the prediction error criteria in the literature. Once we have estimated

c_j's, the NNG estimates are obtained as $\hat{\beta}_j^{NNG} = \hat{c}_j\hat{\beta}_j$. One of the desirable properties of NNG method is that the coefficient estimates do not depend on the scale. Hence, there is no need to scale the variables prior to the analysis. In addition, NNG estimation does subset selection as well. To understand how it does subset selection, let's consider the orthonormal design case where we can estimate c_j as

$$\hat{c}_j = \left(1 - \frac{\lambda}{2\hat{\beta}_j^2}\right)^+. \tag{2.27}$$

Since ${\beta_j}^{NNG} = \hat{c}_j\hat{\beta}_j$, we have $\hat{c}_j = {\beta_j}^{NNG}/\hat{\beta}_j$ where $\hat{\beta}_j$ is the j^{th} OLS estimate. Without loss of generality, we will assume that $\beta_0 = 0$. Then, the NNG problem can be written as the minimizer of

$$Q = \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\frac{\tilde{\beta}_j}{\hat{\beta}_j}. \tag{2.28}$$

Let $\boldsymbol{b}$ be the vector of reciprocals of OLS estimates, that is $\boldsymbol{b} = \{1/\hat{\beta}_j\}_{j=1}^p$. Then we can write (2.28) as

$$\begin{aligned} Q(\boldsymbol{\beta}, \lambda) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{b}^\top\boldsymbol{\beta}, \\ &= \boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta} + \lambda\boldsymbol{b}^\top\boldsymbol{\beta}. \end{aligned}$$

Taking the first derivative,

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^\top\boldsymbol{y} + 2\boldsymbol{X}^\top\boldsymbol{X}\beta + 2\lambda\boldsymbol{b}.$$

Since $Q(\boldsymbol{\beta}, \lambda)$ is convex, there exist a unique minimum at $\boldsymbol{\beta}^{NNG}$. Then

$$\left.\frac{\partial Q}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{NNG}} = 0.$$

By solving the above, we can obtain

$$\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta}^{NNG} = \boldsymbol{X}^\top\boldsymbol{y} - \frac{\lambda}{2}\boldsymbol{b},$$

$$\boldsymbol{\beta}^{NNG} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}(\boldsymbol{X}^\top\boldsymbol{y} - \frac{\lambda}{2}\boldsymbol{b}).$$

Under the orthonormal assumption, $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{I}_p$ and $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\top\boldsymbol{y}$. Then

$$\boldsymbol{\beta}^{NNG} = \hat{\boldsymbol{\beta}} - \frac{\lambda}{2}\boldsymbol{b}.$$

Let ${\beta_j}^{NNG}$ be the j^{th} NNG estimate. Then we have

$$\begin{aligned} {\beta_j}^{NNG} &= \hat{\beta}_j - \tfrac{\lambda}{2\hat{\beta}_j}, \\ &= \left(1 - \tfrac{\lambda}{2\hat{\beta}_j^2}\right)\hat{\beta}_j. \end{aligned}$$

Since we assumed that $c_j > 0$, $\hat{\beta}_j$ and ${\beta_j}^{NNG}$ should take the same sign, or otherwise ${\beta_j}^{NNG} = 0$. Hence, we have the NNG estimate as

$$ {\beta_j}^{NNG} = \left(1 - \frac{\lambda}{2\hat{\beta}_j^2}\right)^+ \hat{\beta}_j. $$

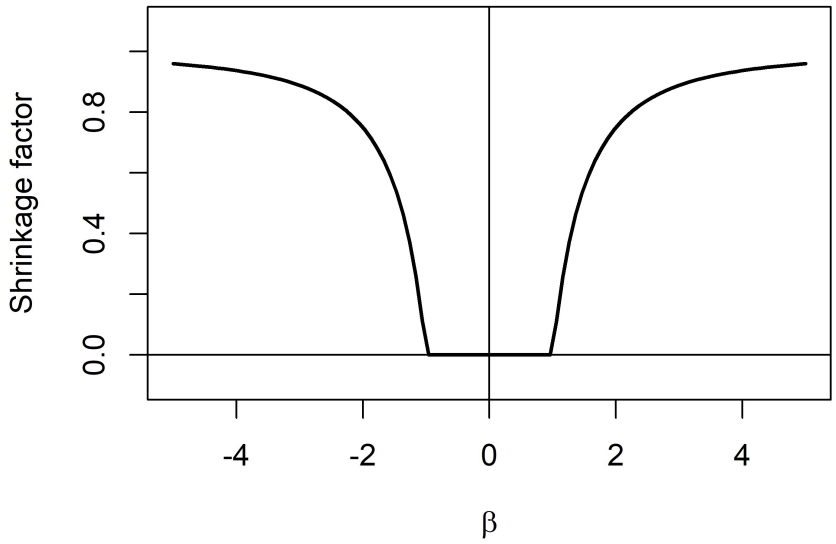


Figure 2.11: Shrinkage factor of NNG estimator for orthonormal design when $\lambda = 2$.

As we observed, NNG method does the subset selection by truncating the parameter estimates when $\hat{\beta}_j^2 < \lambda/2$. The idea can be further explained by plotting the shrinkage factor against $\hat{\beta}_j$. The plot of c_j is presented in Figure 2.11 when $\lambda = 2$ under the orthonormal design assumption. For $\hat{\beta}_j < 1$, the NNG estimate is zero.

For the non-orthonormal case, we cannot obtain a closed form solution. However, the problem can be solved numerically. Breiman (1995) used a modified version of non-negative least squares algorithm given by Lawson and Hanson (1974).

When we are dealing with shrinkage methods, we should note that these methods are less flexible than least squares regression since one needs to restrict the parameter space to a much smaller subspace. However, shrinkage methods such as the lasso and NNG have the additional advantage compared to ridge regression in terms of the interpretability since they produce much simpler models by selecting a subset of predictors to predict the response.

It is worth mentioning that the NNG is almost identical to the adaptive lasso. Consider adaptive lasso

problem in (2.3.1) and let $\gamma = 1$ and $\hat{w}_j = 1/|\hat{\beta}_j|$. Then the adaptive lasso problem is written as

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j|} \right\}. \tag{2.29}$$

Since $|\beta_j^{\text{NNG}}|/|\hat{\beta}_j| = c_j$. This is the same as the NNG estimation with the additional sign constraint $\hat{\beta}_j\beta_j > 0$ (Zou, 2006).

2.6 Model Selection and Prediction Error

To decide the best model and to evaluate the accuracy of our fitted model, we need some measures to quantify the performance of each model. In all shrinkage methods, we have to decide a suitable value for tuning parameter λ, as each corresponds to a new model. One can use multiple approaches such as the lasso, the elastic net and/or ridge regression to fit several models and then, select the best model among them. In this section, we present some of the model evaluation techniques which are widely used to estimate the prediction error of the fitted models.

2.6.1 The Training Set-Testing Set Approach

One of the simplest methods for evaluating the prediction error of a given regression model is to randomly split the dataset of size n into two parts as the training and testing set of sizes $n - m$ and m, respectively. Then we can fit the working model with the training dataset and use the fitted model to predict the response values of the testing set. The prediction error is calculated with the mean squared error

$$\text{MSE}(\tilde{\boldsymbol{\beta}}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 , \tag{2.30}$$

where $\hat{y}_i$ is the predicted response for the i^{th} observation in the testing set and m is the number of observations in the testing set. One of the main disadvantages of this method is that we have less number of observations to train the model. This is not statistically sound unless the sample size is relatively large. When the sample size is small, the estimate of the prediction error will have a higher variance since the model can change significantly depending on which observations falls into which set. In practice, this method usually overestimates the prediction error (James et al., 2013).

2.6.2 Leave-One-Out Cross Validation (LOOCV)

In this approach, we train the model omitting one observation and then we predict the response of the omitted observation. We can calculate the error as $\text{MSE}_i = (y_i - \hat{y}_i)^2$ for the i^{th} omitted observation. Similarly, we can do this for $i = 1, 2, ..., n$ and obtain the LOOCV estimate as

$$\text{CV}_n = \frac{1}{n}\sum_{i=1}^{n}\text{MSE}_i. \tag{2.31}$$

One of the advantages of this approach is that now we use almost all the data to train the model in each step and thus we reduce the bias of the error estimate. Also, now we have less overestimated prediction error than the previous approach (James et al., 2013). However, since we just omit one observation in each step, the splits are not random. Hence, we have almost the same model in each step and we will not have any randomness among the models. This will lead to a prediction error with high variance (Hastie et al., 2009). Another disadvantage of this method is that this approach is computationally intensive as the sample size increases. However, for linear models, there is an equivalent expression for CV_n which is computationally very efficient. To be more specific, for any linear model, one can show that (2.31) can be written as

$$\text{CV}_n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2, \tag{2.32}$$

where h_i, called the leverage of the i^{th} observation, is simply the i^{th} diagonal element of the hat matrix $H = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$. Note that (2.32) only requires the underlying model to be estimated once. For ridge regression, in usual notation $h_i = \boldsymbol{x}_i^\top(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}_p)^{-1}\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ is the vector of predictor values for the i^{th} observation. Usually, $\bar{h} = \text{Trace}(\boldsymbol{X}^\top(\boldsymbol{X}^\top\boldsymbol{X} + k\boldsymbol{I}_p)^{-1}\boldsymbol{X})/n$ is a good approximation for h_i (Breiman, 1995). So, one can further simplify CV_n in (2.32) by replacing $1 - h_i$ with $(1 - \bar{h})$ to obtain a generalized CV as follow

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{(1 - \bar{h})^2}. \tag{2.33}$$

GCV approach is recommended for ridge regression and it is not recommended for subset selection in Breiman (1995).

2.6.3 K-fold Cross Validation

K-fold cross-validation is the most widely used method for calculating the prediction error to select the best tuning parameter in many shrinkage methods including ridge regression, the lasso, the elastic net and NNG. Throughout the thesis, cross-validation technique will be used to evaluate our suggested modeling approaches.

K-fold cross-validation starts with randomly splitting the dataset into K groups of approximately the same size. Similar to LOOCV, to estimate the prediction error one needs to omit the i^{th} fold and train the model with the other $K-1$ folds. The fitted model will then be used to predict the values of the response variable for i^{th} fold and calculate the mean squared error, MSE_i. Similarly, we can repeat this process K times by omitting each fold and calculating the K-fold cross-validation error as

$$\text{CV}_K = \frac{1}{K}\sum_{i=1}^{K}\text{MSE}_i. \tag{2.34}$$

When $K = n$, CV_K reduces to LOOCV. To choose K, we have to find a trade-off between the variance and the bias. Large values of K result in high computational cost and high variance in prediction error estimate. On the other hand, using a small K results in highly biased prediction error estimate with less variance. The common practice is to choose $K = 5$ or $K = 10$. In this thesis, we use 10-fold cross-validation to estimate the prediction error and to estimate the tuning parameters of our suggested models.

2.7 Example 1: The Boston Housing Dataset

This is a famous dataset which is readily available in the MASS library in R. The dataset consists of housing information in suburbs of Boston in 1970, which has been first cited in Harrison and Rubinfeld (1978). The dataset contains 506 observations with 14 attributes namely,

- crim: per capita crime rate by town.
- zn: proportion of residential land zoned for lots over 25,000 sq.ft.
- indus: proportion of non-retail business acres per town.
- chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- nox: nitrogen oxides concentration (parts per 10 million).

- rm: average number of rooms per dwelling.
- age: proportion of owner-occupied units built prior to 1940.
- dis: weighted mean of distances to five Boston employment centers.
- rad: index of accessibility to radial highways.
- tax: full-value property-tax rate per $10,000.
- ptratio: pupil-teacher ratio by town.
- black: $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town.
- lstat: percentage of lower status of the population.
- medv: median value of owner-occupied homes in $1000s.

We will be using the Boston housing dataset throughout this thesis. Suppose we want to build a model to predict medv with other variables as predictors. Figure 2.12 presents the bivariate plots and correlations between a subset of variables. By looking at the plots, we notice that most of the predictors are correlated. We can confirm this by observing the correlations values. We see that the correlation coefficients between the variables indus, nox, age, and dis indicate strong correlations among them.

Now, we implement the aforementioned shrinkage methods for this dataset. As we mentioned, our goal is to build a model to predict medv with all the other variables as predictors. To obtain the lasso, the ridge and the elastic net solutions we used the glmnet library in R. For non-negative garrote, we used lqa library in R. 10-fold cross-validation was used to select the best tuning parameters.

First, let's go through the ridge regression results. Figure 2.13a presents the solution path for ridge regression. The solution path plot (trace plot) is referred to the plot that visualizes the solutions for the ridge regression problem over a grid of λ values. As we observe, increasing the tuning parameter λ, results in all the coefficient estimates to shrink towards zero. However, ridge regression does not set any coefficient exactly to zero. Figure 2.13b shows the cross-validation errors over a grid of λ values. The error bars represent the standard deviation of the prediction error at each λ. We can see that the prediction error goes through a minimum before increasing. Hence, by selecting λ which corresponds to the minimum prediction error, we can obtain a better model than the OLS model in terms of the prediction accuracy and the stability of coefficient estimates.

Solution path for the lasso has been presented in Figure 2.14a. As we increase the tuning parameter λ, all the coefficient estimates shrink towards zero. However, unlike ridge regression, after a certain λ, the lasso produces sparse models by setting some coefficients exactly to zero. For a very large value of λ, all

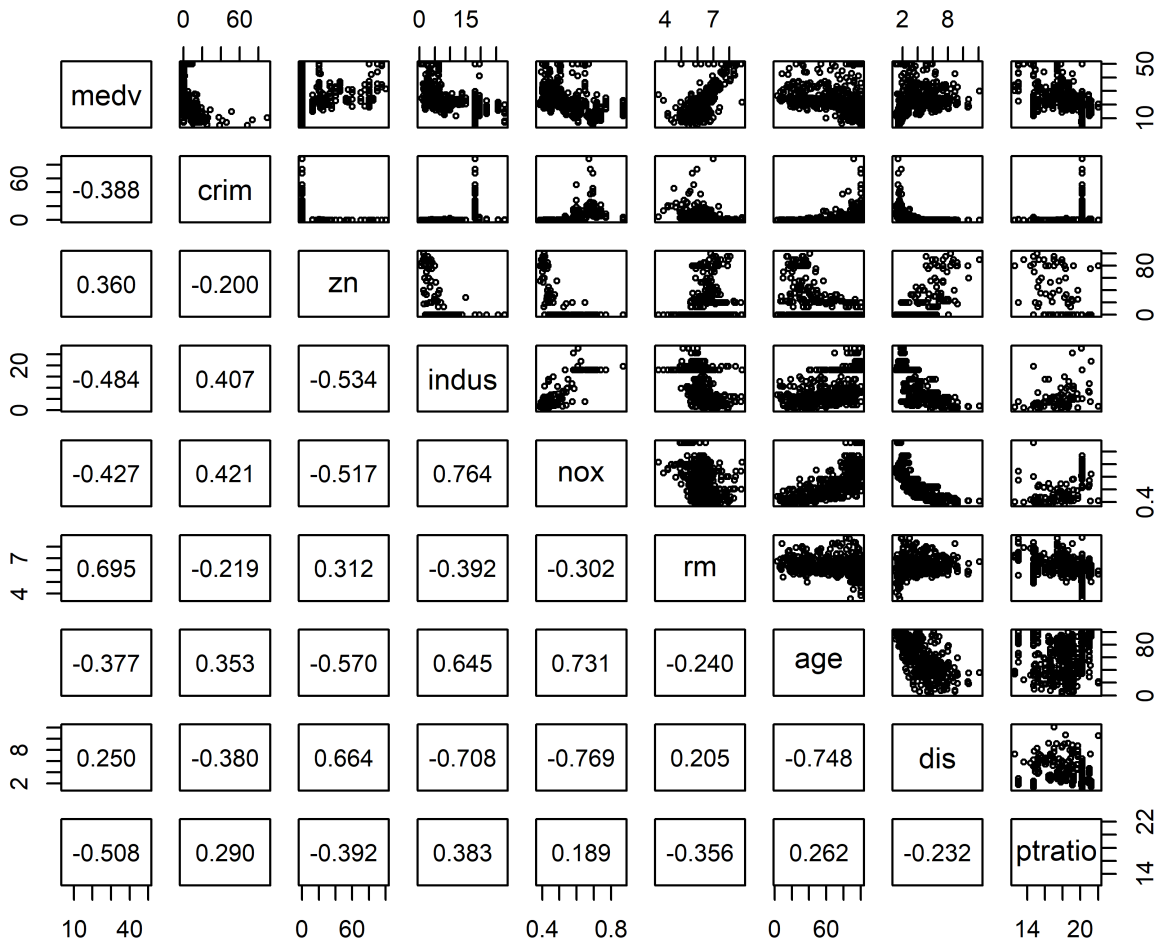
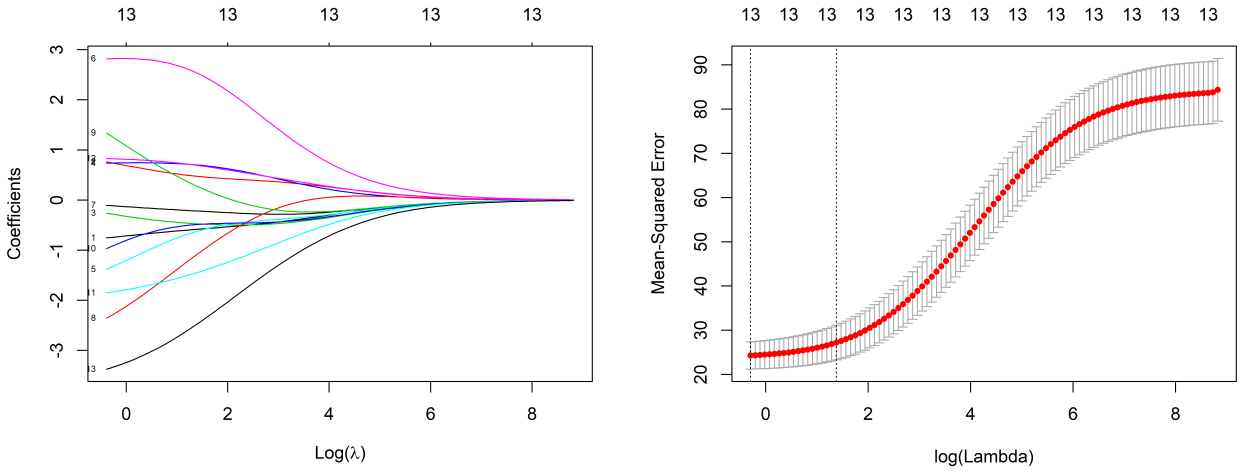


Figure 2.12: Bivariate plots and bivariate correlations of variables of the Boston housing dataset.

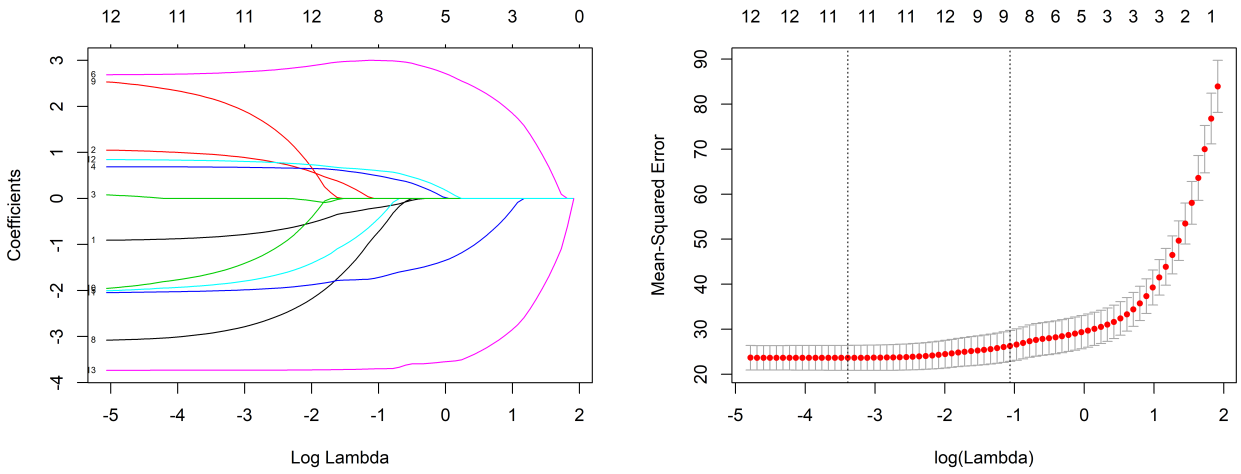
the coefficients will set to be exactly zero. Figure 2.14b shows the cross-validation errors against λ. As we increase λ from zero, cross-validation error goes through a minimum for the lasso as well. Hence, by selecting λ which corresponds to the minimum prediction error, using the lasso estimator we can obtain a much simpler model with a better prediction accuracy than the OLS model. In practice, we sometimes use the maximum λ within one standard deviation units from the point corresponding to the minimum cross-validation error. This is called the one standard error rule (James et al., 2013). Selecting the best λ with one standard error rule results in further simplified model with more number of zero coefficients than the model which has the minimum cross-validation error.

Solution path for the elastic net with $\alpha = 0.5$ is presented in Figure 2.15a. Elastic net also behaves



(a) (b)

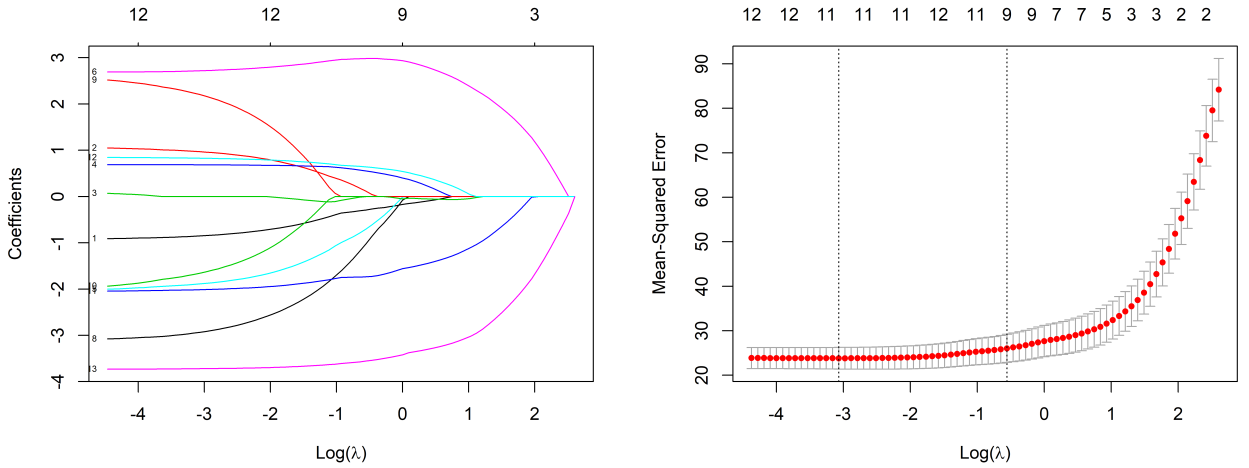
Figure 2.13: (a) Coefficients of the multiple linear regression model under the ridge regression approach for the Boston housing dataset. (b) Cross validation error of ridge regression for the Boston housing dataset.



(a) (b)

Figure 2.14: (a) Coefficients of the multiple linear regression model under the lasso approach for the Boston housing dataset. (b) Cross validation error of the lasso for the Boston housing dataset.

very similar to the lasso and, as we increase the tuning parameter λ, all the coefficient estimates shrink towards zero setting some coefficients exactly to zero. Figure 2.15b shows the cross-validation errors against λ. Similar to the above cases, as we increase λ, cross-validation error goes through a minimum.



(a) (b)

Figure 2.15: (a) Coefficients of the multiple linear regression model under the elastic net with $\alpha = 0.5$ for the Boston housing dataset. (b) Cross validation error of the elastic net with $\alpha = 0.5$ for the the Boston housing dataset.

Table 2.1 presents the summary of the estimated models with each of the shrinkage approach along with the OLS approach. All the shrinkage models were obtained with the tuning parameter which minimizes the cross-validation error. Coefficient estimates with ridge regression are much smaller than the least squares estimates. At the best value of λ, the lasso uses only eight predictors in the model. Elastic net also uses only eight or nine predictors at different α values. The most important observation is that the elastic net and the lasso produce very simple models by reducing total of 13 predictors to a lower number of predictors. As for non-negative garrote, only two coefficients have been set exactly to zero at the best value of λ. This example will continue with some new shrinkage methods which we propose in the proceeding chapters.

Table 2.1: Coefficient estimates and prediction errors with different shrinkage methods for Boston housing dataset.

	OLS	Ridge	Lasso	EN($\alpha = 0.5$)	EN($\alpha = 0.8$)	NNG
crim	-0.929	-0.584	-0.206	-0.290	-0.161	-1.078
zn	1.083	0.476	0	0.117	0	0.962
indus	0.141	-0.472	0	0	0	0
chas	0.682	0.702	0.507	0.553	0.428	0.696
nox	-2.059	-0.627	-0.520	-0.706	-0.105	-2.057
rm	2.677	2.541	3.002	2.982	2.980	2.653
age	0.019	-0.217	0	0	0	0
dis	-3.107	-1.093	-0.821	-1.027	-0.298	-3.123
rad	2.665	0.284	0	0	0	2.484
tax	-2.079	-0.488	0	0	0	-1.751
ptratio	-2.063	-1.451	-1.735	-1.734	-1.628	-2.095
black	0.850	0.695	0.611	0.641	0.569	0.749
lstat	-3.747	-2.484	-3.703	-3.556	-3.626	-3.894
MSE	23.918	24.197	23.914	23.902	23.899	24.397

Chapter 3

Quadratic Garrote

In this chapter, we study two shrinkage approaches which can be used to obtain an asymmetric shrinkage on coefficient estimates. First, we study the quadratic garrote suggested in Breiman (1995) which can be used to shrink the coefficient estimates inversely proportional to their size. Then, we propose the generalized quadratic garrote method as an extension of the quadratic garrote. We find the generalized quadratic garrote approach to be more practically sound and the user to have more control over the amount of shrinkage on each regression coefficient estimate. Furthermore, we study the properties of the suggested estimators theoretically, and using simulation studies. Finally, we illustrate the practical use of suggested shrinkage approaches with the Boston Housing Dataset.

3.1 Introduction

Breiman (1995) presents the non-negative garrote as a better subset selection method. In the non-negative garrote, we estimated $\{c_j\}_{j=1}^p$ to minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} c_j\hat{\beta}_j x_{ij})^2 + \lambda\sum_{j=1}^{p} c_j.$$

Non-negative garrote approach imposes the penalty $\sum_{j=1}^p c_j$, on SSE where we consider $\{c_j\}$ to be positive. By using $\sum_{j=1}^p c_j^2$ as the penalty, we can extend the idea of the non-negative garrote to obtain a scale-invariant substitute for ridge regression. Consider the optimization problem of estimating $\{c_j\}_{j=1}^p$ to minimize

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij})^2 \text{ subject to } \sum_{j=1}^{p} c_j^2 \leq s, \tag{3.1}$$

where, $s > 0$ and $\hat{\beta}_j$'s are the OLS estimates. For future reference, we name this approach to be the quadratic garrote (QG). Then the j^{th} QG estimator is obtained as $\hat{\beta}_j{}^{Q} = c_j \hat{\beta}_j$. Equivalently, we can re-write the problem using the Lagrange multiplier as

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij})^2 + \lambda \sum_{j=1}^{p} c_j^2, \tag{3.2}$$

where, λ is the tuning parameter. Even though Breiman (1995) suggested the approach, they did not implement the idea. Nevertheless, the author expects quadratic garrote to be uniformly more accurate than ridge regression and to be almost as stable as the ridge regression. We remark that there is no other work in the literature which addressed the quadratic garrote problem. Hence, we implement the quadratic garrote in this chapter along with its theoretical framework.

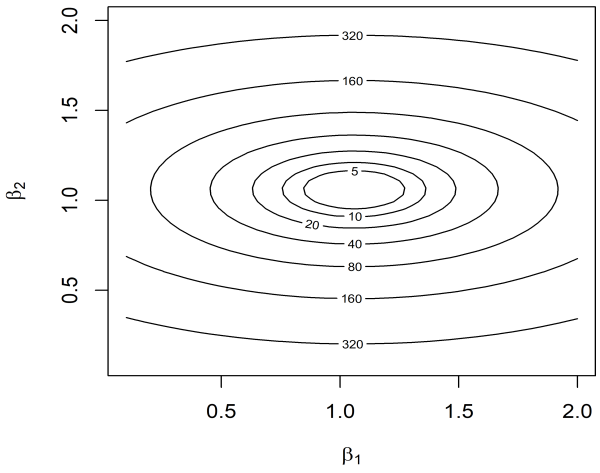


Figure 3.1: QG constraints for different s values. Each circle represents those $(\beta_1, \beta_2) \in \mathbb{R}$ such that $\beta_1^2/\hat{\beta}_1^2 + \beta_2^2/\hat{\beta}_2^2 = s, s \in \{5, 10, 20, 40, \ldots\}$ where $\hat{\boldsymbol{\beta}} = (1, 2)^\top$.

For the two predictor case, the nature of the QG penalty can be seen in Figure 3.1 considering $\hat{\boldsymbol{\beta}} = (1, 2)^\top$. The QG penalty is elliptical in shape. Hence, unlike the ridge penalty, it shrinks the two OLS coefficient estimate differently. In this case, QG method shrinks β_2 two times than β_1. As we observed in the other

shrinkage methods, QG approach also restricts the parameter space of the coefficients by imposing the QG penalty and then selects the best estimates in the restricted space. Consider the same example data with two predictors X_1 and X_2, which we presented in Figure 2.2 in Section 2.2. Figure 3.2 shows the solution for the QG garrote estimate for the example dataset along with the QG penalty at $s = 0.5$ and the contours of SSE. When we compare the QG solution with the ridge solution on Figure 2.3 where the penalty is a circle, we see that using the quadratic garrote, we the larger coefficient ($\hat{\beta}_1$) has been shrunk less than the smaller one.

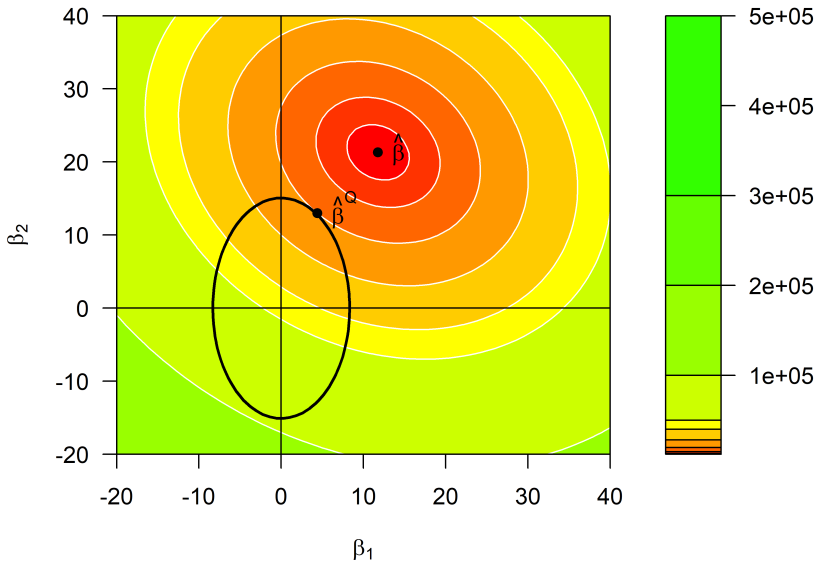


Figure 3.2: Contour plot of SSE with the QG penalty and the QG solution at $s = 0.5$.

We can further generalize the quadratic garrote to obtain a more flexible shrinkage approach which has a wide range of practical applications. Following theorem defines the generalized quadratic garrote estimation problem.

Theorem 1 *Consider the multiple linear regression model* $y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i; i = 1, \ldots, n$, *where* ϵ_i *are iid with* $E(\epsilon_i) = 0$ *and* $V(\epsilon_i) = \sigma^2$. *Let* $\tilde{\boldsymbol{\beta}}$ *be the generalized quadratic garrote estimate given by*

$$\tilde{\boldsymbol{\beta}} = \underset{\beta_0, \beta_1, \ldots, \beta_p}{argmin} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p} d_j^2 \beta_j^2 \leq s, \tag{3.3}$$

where, d_j^2*'s are some positive quantities (shrinking factors) which can depend on* $\hat{\beta}_j$ *or they can be fixed constants. Then, in usual matrix notation, the solution for* (3.3) *is obtained as*

$$\tilde{\boldsymbol{\beta}}(\lambda) = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} \boldsymbol{X}^\top \boldsymbol{y},$$

where $\boldsymbol{B}$ *is a* $p \times p$ *diagonal matrix with diagonal elements* d_j^2*, and* λ *is determined such that* $\sum_{j=1}^{p} d_j^2 \beta_j^2 = s$*.*

We notice that the minimization problem in (3.3) has some similarity with the generalized ridge regression approach suggested in Hoerl and Kennard (1970). However, there are some major differences between the two approaches. We can rewrite minimization problem in (3.3) with the Lagrange multiplier as

$$Q(\boldsymbol{\beta}, \lambda) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} d_j^2 \beta_j^2. \tag{3.4}$$

Here, $\lambda \sum_{j=1}^{p} d_j^2 \beta_j^2$ is the penalty term. For the comparison purpose, assume d_j^2s to be unknown and let $\lambda d_j^2 = \lambda_j$. Then we have

$$Q(\boldsymbol{\beta}, \lambda) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \sum_{j=1}^{p} \lambda_j \beta_j^2, \tag{3.5}$$

which is the generalized ridge regression problem.

Since we assume d_j^2 to be known in the generalized garrote approach, the size of the penalty is defined with the single tuning parameter λ. Contrary, in the generalized there are p number of tuning parameters. Hoerl and Kennard (1970) defined the generalized ridge regression in the eigenvector space. Hence, prior to applying the generalized ridge method, we have to project the design matrix into the eigen space. As a result, the tuning parameter λ_j does not corresponds to the level of shrinkage on j^{th} coefficient estimate. However, in our approach, λd_j^2 directly defines the amount of shrinkage on the corresponding coefficient estimate.

To prove the Theorem 1, consider the optimization problem in 3.4. Without loss of generality, we can assume $\boldsymbol{y}$ has been centered and $\beta_0 = 0$. Let $\boldsymbol{X}$ be the centered design matrix with the dimensions $n \times p$. By rewriting (3.4) in the vector notation we have

$$\begin{aligned} Q(\boldsymbol{\beta}, \lambda) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{B} \boldsymbol{\beta}, \\ &= \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda \boldsymbol{\beta}^\top \boldsymbol{B}\boldsymbol{\beta}. \end{aligned}$$

Taking the first derivative of $Q(\boldsymbol{\beta}, \lambda)$ with respect to $\boldsymbol{B}$ we have

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + 2\lambda \boldsymbol{B}\boldsymbol{\beta}.$$

Since $Q(\boldsymbol{\beta}, \lambda)$ is convex, there exists a unique minimum at $\tilde{\boldsymbol{\beta}}$, such that

$$\left.\frac{\partial Q}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = 0. \tag{3.6}$$

By solving (3.6), we obtain

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})\tilde{\boldsymbol{\beta}} = \boldsymbol{X}^\top \boldsymbol{y}.$$

By further simplifying, we have

$$\tilde{\boldsymbol{\beta}}(\lambda) = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

We can easily derive the ridge estimator and the quadratic garrote estimator with Theorem 1.

Example 1: Ridge regression

Setting all d_j^2's to be 1 in (3.4), we have the classical ridge regression problem. Then, $\boldsymbol{B}$ is the $p \times p$ identity matrix and we have the solution, $\hat{\boldsymbol{\beta}}^R = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$.

Example 2: Quadratic garrote

The quadratic garrote minimization problem in 3.3 can be rewritten with the Lagrange multiplier as

$$Q(c_1, c_2, \ldots, c_p, \lambda) = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{k} c_j^2. \tag{3.7}$$

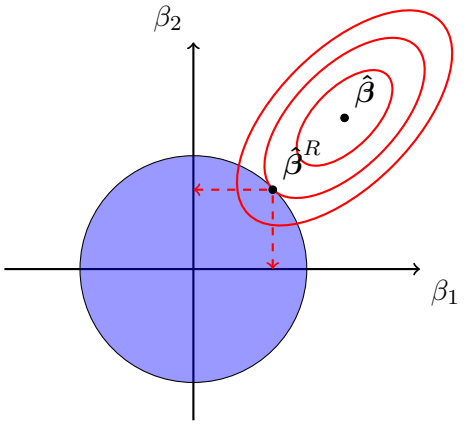
Let $\hat{\beta}_j{}^Q = c_j \hat{\beta}_j$ be the new quadratic garrote coefficients with $c_j = \hat{\beta}_j^Q / \hat{\beta}_j$. Substituting c_j in (3.7) we obtain the quadratic garrote estimates as

$$\hat{\boldsymbol{\beta}}^Q = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \left(\frac{1}{\hat{\beta}_j^2}\right) \beta_j^2. \tag{3.8}$$

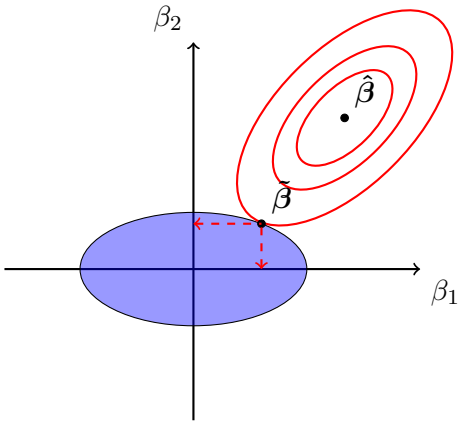
Applying Theorem 1 with $d_j^2 = 1/\hat{\beta}_j{}^2$ we derive the vector of the quadratic garrote estimator as

$$\hat{\boldsymbol{\beta}}^{(Q)}(\lambda) = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

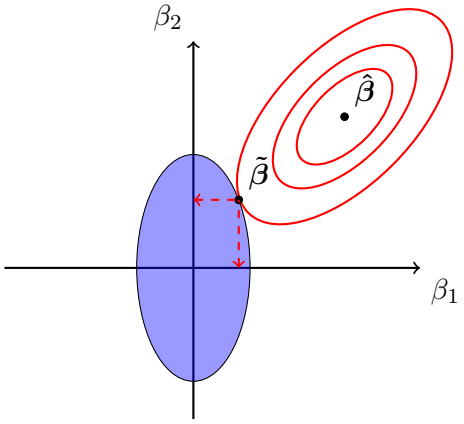
where $\boldsymbol{B} = \operatorname{diag}(1/\hat{\beta}_j{}^2)$ and $\hat{\beta}_j$'s are the OLS estimators.



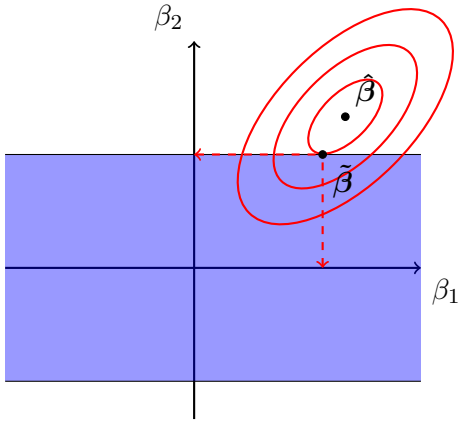
(a) Ridge penalty: $\beta_1^2 + \beta_2^2 < t$; i.e. $d_1^2 = d_2^2 = 1$



(b) $\beta_1^2 + 2\beta_2^2 < t$; i.e. $d_1^2 < d_2^2$



(c) $2\beta_1^2 + \beta_2^2 < t$; i.e. $d_1^2 > d_2^2$



(d) No penalty on β_1: $\beta_2^2 < t$; i.e. $d_1^2 = 0$

Figure 3.3: Nature of the penalty on parameters with different d_j^2s for two predictor case.

3.2 Practical Importance of the Generalized Quadratic Garrote

The generalized quadratic garrote estimator posses a practical advantage over other shrinkage methods. In the preceding sections, we saw that the ridge regression and the lasso do the shrinkage for all the parameters similarly. Sometimes, one might not want to shrink all the coefficients similarly. Imagine one wants to keep a specific set of variables unshrunk or with minimum shrinking while applying more shrinkage on some other variables. With the quadratic garrote, we can arbitrarily decide the level of shrinkage on each variable while maintaining a reduction in MSE. For two variables case, Figure 3.3 illustrates the idea visually. Contours represent the SSE of OLS estimation and the minimum SSE is achieved at $\hat{\boldsymbol{\beta}}$. Shaded regions represent the generalized quadratic garrote constraints for different d_j vectors. We see that the nature of the constraint changes with d_j. As we see in Figure 3.3 (a), when

each $d_j = 1$, the constraint is the same as the ridge constraint, and we get with ridge regression solution. Observe, Figure 3.3 (b), where $d_1^2 < d_2^2$. In this case, β_2 is twice constrained than β_1. Hence, estimated coefficient for β_2 is shrunken more towards zero than β_1. This setting is good if we want less shrinkage on β_1. This story is the opposite for Figure 3.3 (c). Figure 3.3 (d) shows the case when there is no penalty on β_1. In this case, only β_2 will be constrained and β_1 can take any value. In this specific example, the estimate of β_1 is less than the OLS estimate. However, this is not always true. $\hat{\beta}_1^Q$ can be even larger than the corresponding OLS estimate since it is not constrained.

3.3 Variance and Bias of Generalized Quadratic Garrote Estimator

Recall the generalized quadratic garrote estimator we presented in Theorem 1, $\tilde{\boldsymbol{\beta}}(\lambda) = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$. We can obtain the relationship between $\tilde{\boldsymbol{\beta}}$ and OLS estimator $\hat{\boldsymbol{\beta}}$ as

$$\begin{aligned}
\tilde{\boldsymbol{\beta}} &= (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}, \\
&= (\boldsymbol{I}_p + \lambda \boldsymbol{B} (\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1} (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}, \\
&= (\boldsymbol{I}_p + \lambda \boldsymbol{B} (\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1} \hat{\boldsymbol{\beta}}, \\
&= \boldsymbol{Z} \hat{\boldsymbol{\beta}}. \qquad (3.9)
\end{aligned}$$

Let $\tilde{\boldsymbol{\beta}} = \boldsymbol{W} \boldsymbol{X}^\top \boldsymbol{Y}$ where $\boldsymbol{W} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1}$. We can show that

$$\boldsymbol{Z} = \boldsymbol{I}_p - \lambda \boldsymbol{B} (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{B})^{-1} = \boldsymbol{I}_p - \lambda \boldsymbol{B} \boldsymbol{W}. \qquad (3.10)$$

This can be easily obtained by applying Woodbury matrix identity which was suggested in Henderson and Searle (1981). We use a simplified version of Woodbury matrix identity which can be found in Hager (1989). Suppose both $\boldsymbol{A}$ and $\boldsymbol{I} - \boldsymbol{V} \boldsymbol{A}^{-1} \boldsymbol{U}$ are invertible, then $\boldsymbol{A} - \boldsymbol{U} \boldsymbol{V}$ is invertible and its inverse can be written as

$$[\boldsymbol{A} - \boldsymbol{U} \boldsymbol{V}]^{-1} = \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1} \boldsymbol{U} (\boldsymbol{I} - \boldsymbol{V} \boldsymbol{A}^{-1} \boldsymbol{U})^{-1} \boldsymbol{V} \boldsymbol{A}^{-1}.$$

Using $\boldsymbol{A} = \boldsymbol{I}_p, U = \lambda \boldsymbol{B}$, and $\boldsymbol{V} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}$, on $\boldsymbol{Z} = (\boldsymbol{I}_p + \lambda \boldsymbol{B} (\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}$ and further simplification, we can derive (3.10). The results which we derived in (3.9) and (3.10) are useful in the proceeding section.

Let's consider the two cases; $\boldsymbol{B}$ is independent of $\boldsymbol{Y}$ and, $\boldsymbol{B}$ is not independent of $\boldsymbol{Y}$. In the first case, it is easy to obtain expressions for the variance and the bias of $\tilde{\boldsymbol{\beta}}$. However, in the second case, it is

not straight forward to obtain exact expressions for the variance and bias of $\tilde{\boldsymbol{\beta}}$. Instead, we derive approximate solutions for them.

3.3.1 Case 1: B is Independent of $\boldsymbol{y}$

When $\boldsymbol{B}$ is independent of $\boldsymbol{y}$, we can easily obtain the expectation and variance of $\tilde{\boldsymbol{\beta}} = \boldsymbol{Z}\hat{\boldsymbol{\beta}}$. Since $\hat{\boldsymbol{\beta}}$ is unbiased for $\boldsymbol{\beta}$,

$$E(\tilde{\boldsymbol{\beta}}) = \boldsymbol{Z}\boldsymbol{\beta}, \tag{3.11}$$

and

$$Var(\tilde{\boldsymbol{\beta}}) = \boldsymbol{Z}Var(\boldsymbol{\beta})\boldsymbol{Z}^\top = \sigma^2\boldsymbol{Z}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top. \tag{3.12}$$

It is easy to see that the quadratic garrote estimator is biased ($E(\tilde{\boldsymbol{\beta}}) \neq \boldsymbol{\beta}$). Consider the mean squared error of $\tilde{\boldsymbol{\beta}}$ given by

$$\text{MSE}(\tilde{\boldsymbol{\beta}}) = E\left[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})\right]. \tag{3.13}$$

This can be further decomposed as

$$\begin{aligned}
\text{MSE}(\tilde{\boldsymbol{\beta}}) &= E\left[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})\right], \\
&= E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right], \\
&= E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})^\top(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})\right], \\
&= E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta})^\top(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta})\right] + E\left[(\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})^\top(\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})\right] + 2E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta})^\top(\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})\right], \\
&= E\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top\boldsymbol{Z}^\top\boldsymbol{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right] + (\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta})^\top(\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta}) + 2E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta})^\top\right](\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\beta}). \qquad (3.14)
\end{aligned}$$

Since $E\left[(\boldsymbol{Z}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\boldsymbol{\beta})^\top\right] = 0$,

$$\text{MSE}(\tilde{\boldsymbol{\beta}}) = E\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top\boldsymbol{Z}^\top\boldsymbol{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right] + \boldsymbol{\beta}^\top(\boldsymbol{Z} - \boldsymbol{I}_p)^\top(\boldsymbol{Z} - \boldsymbol{I}_p)\boldsymbol{\beta}. \tag{3.15}$$

Now, $(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top\boldsymbol{Z}^\top\boldsymbol{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ is a quadratic function of $\hat{\boldsymbol{\beta}}$. Using the fact that $E(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = 0$, we can further simplify the expectation as

$$\text{MSE}(\tilde{\boldsymbol{\beta}}) = \sigma^2\text{Trace}[(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top\boldsymbol{Z}] + \boldsymbol{\beta}^\top(\boldsymbol{Z} - \boldsymbol{I}_p)^\top(\boldsymbol{Z} - \boldsymbol{I}_p)\boldsymbol{\beta}. \tag{3.16}$$

Consider the matrix $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top\boldsymbol{Z}$. We can further simplify this by expanding $\boldsymbol{Z}$ with it's original form $\boldsymbol{Z} = [\boldsymbol{I}_p + \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}]^{-1}$ and, the alternative form $\boldsymbol{Z} = \boldsymbol{I}_p - \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}$ as

$$\begin{aligned}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top\boldsymbol{Z} &= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\left[\boldsymbol{I}_p + \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\right]^{-1}\left[\boldsymbol{I}_p - \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\right],\\ &= (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\left[\boldsymbol{I}_p - \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\right],\\ &= (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1} - \lambda(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}.\end{aligned}$$

Hence,

$$\begin{aligned}&\text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top\boldsymbol{Z}\right]\\ &= \text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1} - \lambda(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\right],\\ &= \text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\right] - \text{Trace}\left[\lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-2}\right],\\ &= \text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\right] - \text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B} - \boldsymbol{X}^\top\boldsymbol{X})(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-2}\right],\\ &= \text{Trace}\left[\boldsymbol{X}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-2}\right]. \qquad (3.17)\end{aligned}$$

Now, we simplify the second part of (3.16). Since, $\boldsymbol{Z} = \boldsymbol{I}_p - \lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}$, we have $\boldsymbol{Z} - \boldsymbol{I}_p = -\lambda\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}$. Then,

$$\boldsymbol{\beta}^\top(\boldsymbol{Z} - \boldsymbol{I}_p)^\top(\boldsymbol{Z} - \boldsymbol{I}_p)\boldsymbol{\beta} = \lambda^2\boldsymbol{\beta}^\top(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{\beta}.$$

Finally we have MSE($\tilde{\boldsymbol{\beta}}$) as below

$$\begin{aligned}\text{MSE}(\tilde{\boldsymbol{\beta}}) &= \sigma^2\text{Trace}\left[\boldsymbol{X}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-2}\right] + \lambda^2\boldsymbol{\beta}^\top(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{B}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{B})^{-1}\boldsymbol{\beta},\\ &= \gamma_1(\lambda) + \gamma_2(\lambda). \qquad (3.18)\end{aligned}$$

Here, $\gamma_1(\lambda)$ is the sum of the total variation of $\tilde{\boldsymbol{\beta}}$ vector or simply the sum of the diagonal elements of $Var(\tilde{\boldsymbol{\beta}})$ in (3.12). $\gamma_2(\lambda)$ is the squared bias of $\tilde{\boldsymbol{\beta}}$. It is easy to see that $\gamma_1(\lambda)$ is monotonically decreasing in λ. Furthermore, we can show that

$$\lim_{\lambda\to 0^+}\gamma_1(\lambda) = \sigma^2\text{Trace}\left[(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\right],$$

and

$$\lim_{\lambda\to\infty}\gamma_1(\lambda)=0.$$

Re-writing $\gamma_2(\lambda)$ as

$$\gamma_2(\lambda)=\boldsymbol{\beta}^\top\left(\frac{\boldsymbol{X}^\top\boldsymbol{X}}{\lambda}+\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\top\boldsymbol{B}\left(\frac{\boldsymbol{X}^\top\boldsymbol{X}}{\lambda}+\boldsymbol{B}\right)^{-1}\boldsymbol{\beta},$$

we can see that $\gamma_2(\lambda)$ is a monotonically increasing function in λ. Also, we can show that

$$\lim_{\lambda\to0^+}\gamma_2(\lambda)=0,\qquad\text{and}\qquad\lim_{\lambda\to\infty}\gamma_2(\lambda)=\boldsymbol{\beta}^\top\boldsymbol{\beta}.$$

That is, $\gamma_2(\lambda)$ is bounded above by the squared length of $\boldsymbol{\beta}$. To further study the properties related to these error components, assume that the design matrix $\boldsymbol{X}$ is orthonormal. That is $\boldsymbol{X}^\top\boldsymbol{X}=\boldsymbol{I}_p$. Then,

$$\begin{aligned}\gamma_1^*(\lambda)&=\sigma^2\text{Trace}\left[\boldsymbol{I}_p(\boldsymbol{I}_p+\lambda\boldsymbol{B})^{-2}\right],\\&=\sigma^2\sum_{j=1}^{p}\frac{1}{(1+\lambda d_j^2)^2}.\end{aligned}\tag{3.19}$$

We see that $\gamma_1^*(\lambda)$ is a monotonically decreasing function in λ with

$$\lim_{\lambda\to0^+}\gamma_1^*(\lambda)=\sigma^2p,\ \text{and}\ \lim_{\lambda\to\infty}\gamma_1^*(\lambda)=0.$$

Also,

$$\begin{aligned}\gamma_2^*(\lambda)&=\lambda^2\boldsymbol{\beta}^\top(\boldsymbol{I}_p+\lambda\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{B}(\boldsymbol{I}_p+\lambda\boldsymbol{B})^{-1}\boldsymbol{\beta},\\&=\lambda^2\sum_{j=1}^{p}\frac{d_j^4\beta_j^2}{(1+\lambda d_j^2)^2},\\&=\sum_{j=1}^{p}\frac{d_j^4\beta_j^2}{(\frac{1}{\lambda}+d_j^2)^2}.\end{aligned}\tag{3.20}$$

Hence, $\gamma_2^*(\lambda)$ is a monotonically increasing function in λ. The limits are as below.

$$\lim_{\lambda\to0^+}\gamma_2^*(\lambda)=0,\ \text{and}\ \lim_{\lambda\to\infty}\gamma_2^*(\lambda)=\sum_{i=1}^{p}\beta_j^2=\boldsymbol{\beta}^\top\boldsymbol{\beta}.$$

That is, $\gamma_2^*(\lambda)$ is bounded above by length of $\boldsymbol{\beta}$. Now consider

$$\text{MSE}(\tilde{\boldsymbol{\beta}})^*=\gamma_2^*(\lambda)+\gamma_2^*(\lambda).\tag{3.21}$$

Limits of $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ are obtained as

$$\lim_{\lambda \to 0^+} \text{MSE}(\tilde{\boldsymbol{\beta}})^* = \sigma^2 p, \tag{3.22}$$

which is the $\text{MSE}(\hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is the vector of OLS estimators, and

$$\lim_{\lambda \to \infty} \text{MSE}(\tilde{\boldsymbol{\beta}})^* = \boldsymbol{\beta}^\top \boldsymbol{\beta}. \tag{3.23}$$

The first derivative of (3.21) is obtained as

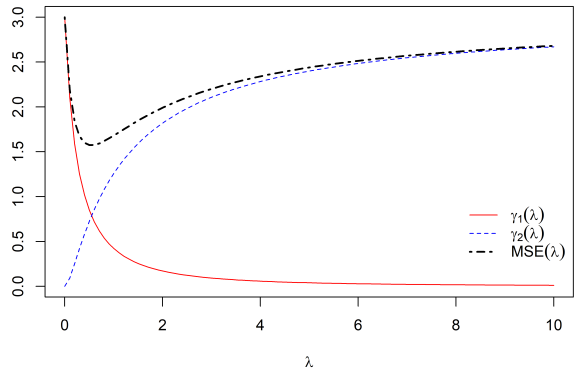
$$\begin{aligned} \frac{\partial}{\partial \lambda} \text{MSE}(\tilde{\boldsymbol{\beta}})^* &= -2\sigma^2 \sum_{j=1}^{p} \frac{d_j^2}{(1+\lambda d_j^2)^3} + 2\lambda \sum_{j=1}^{p} \frac{\beta_j^2 d_j^4}{(1+\lambda d_j^2)^3}, \\ &= 2 \sum_{j=1}^{p} \frac{d_j^2 (\lambda \beta_j^2 d_j^4 - \sigma^2)}{(1+\lambda d_j^2)^3}. \end{aligned} \tag{3.24}$$

The derivative (3.24) can have any sign. Hence, $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ is not a monotone function, and we cannot conclude that it is monotonically increasing or monotonically decreasing as we increase λ. However, we can show that $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ first go through a minimum before it increases. The derivative of $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ will be negative if $d_j^2(\lambda \beta_j^2 d_j^4 - \sigma^2) < 0$, for all $j \in 1, ..., p$, or equivalently $\lambda < \sigma^2/(\beta_j^2 d_j^4)$ for all $j \in 1, ..., p$. We can obtain a upper bound for λ as

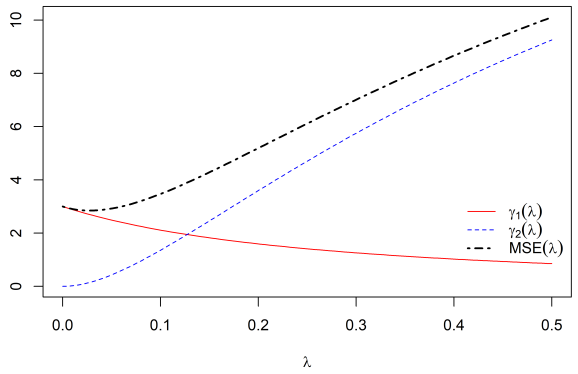
$$\lambda < \sigma^2 / \max(\beta_j^2 d_j^4). \tag{3.25}$$

Hence, $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ will be decreasing for some $\lambda < \sigma^2/max(\beta_j^2 d_j^4)$. So, there exist some $\tilde{\boldsymbol{\beta}}$, such that $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ is lower than $\sigma^2 p = \text{MSE}(\hat{\boldsymbol{\beta}})$.

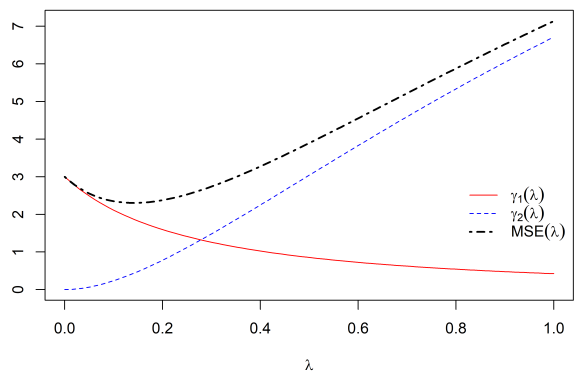
Figures 3.4a to 3.4d show the behavior of these error components for different $\{d_j\}$ vectors and different $\boldsymbol{\beta}$ vectors. In all the examples, we see the expected theoretical behavior for the three curves. Also we notice that, when large coefficients get larger weights, minimum of $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ achieves quickly and we see a drastic drop in $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$. In contrast, when larger coefficients get smaller weights, minimum of $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$ achieves slowly and also we do not observe a significant drop in $\text{MSE}(\tilde{\boldsymbol{\beta}})^*$. This is an indicator that the idea of the quadratic garrote might be a better substitute for the ridge regression since it defines d_j^2's inversely proportional to the squared OLS estimates.



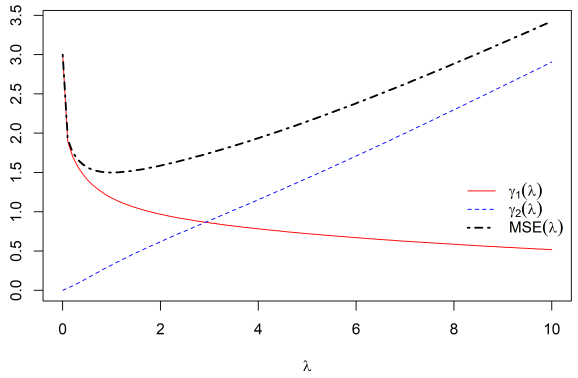
(a) $\boldsymbol{\beta} = (2, 2, 2)^\top$, $\boldsymbol{B} = \text{diag}\{1, 2, 3\}$ and $\sigma = 1$



(b) $\boldsymbol{\beta} = (0.2, 2, 5)^\top$, $\boldsymbol{B} = \text{diag}\{1, 2, 3\}$ and $\sigma = 1$.



(c) $\boldsymbol{\beta} = (5, 2, 0.2)^\top$, $\boldsymbol{B} = \text{diag}\{1, 2, 3\}$ and $\sigma = 1$.



(d) $\boldsymbol{\beta} = (0.2, 2, 5)^\top$, $\boldsymbol{B} = \text{diag}\{1/0.2^2, 1/2^2, 1/5^2\}$ and $\sigma = 1$.

Figure 3.4: MSE of generalized quadratic garrote estimator

3.3.2 Case 2: B Depends on $\boldsymbol{y}$

Consider $\boldsymbol{B} = \text{diag}(d_j^2)$ and $d_j^2 = f_j(Y)$. As we mentioned earlier, since $\boldsymbol{B}$ depends on $\boldsymbol{y}$, we cannot obtain a closed form solution for the expectation and variance of $\tilde{\boldsymbol{\beta}}$. However, by applying the Taylor series expansion for $\boldsymbol{B}$, we can obtain approximate solutions. Let's consider d_j^2 to be a function of $\hat{\beta}_j$. That is $d_j^2 = f_j(\hat{\beta}_j)$. Since $\boldsymbol{B}$ is a diagonal matrix, we can expand each d_j^2 independently around β_j, that is

$$f_j(\hat{\beta}_j) = f_j(\beta_j) + (\hat{\beta}_j - \beta_j)f_j'(\beta_j) + \frac{1}{2}(\hat{\beta}_j - \beta_j)^2 f_j''(\beta_j),$$
$$= f_j(\beta_j) + r_j(\hat{\beta}_j).$$

Then, we can decompose $\boldsymbol{B}$ as

$$\boldsymbol{B} = \boldsymbol{D} + \boldsymbol{R}, \tag{3.26}$$

where $\boldsymbol{D} = \text{diag}(f_j(\beta_j))$ and $R = \text{diag}(r_j(\hat{\beta}_j))$. Using the decomposition in (3.26),

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{I}_p + \lambda \boldsymbol{B}(\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}\hat{\boldsymbol{\beta}},$$
$$= (\boldsymbol{I}_p + \lambda(\boldsymbol{D} + \boldsymbol{R})(\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}\hat{\boldsymbol{\beta}}.$$

Applying the Woodbury matrix identity,

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{I}_p + \lambda \boldsymbol{D}(\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}\hat{\boldsymbol{\beta}} + \boldsymbol{R}^*\hat{\boldsymbol{\beta}},$$
$$\approx (\boldsymbol{I}_p + \lambda \boldsymbol{D}(\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}\hat{\boldsymbol{\beta}}, \tag{3.27}$$

where, $\boldsymbol{R}^* = \lambda \boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{D})^{-1}\boldsymbol{R}\left[\boldsymbol{X}^\top \boldsymbol{X} + \lambda(\boldsymbol{D} + \boldsymbol{R})\right]^{-1}$.

Even if $(\boldsymbol{I}_p + \lambda \boldsymbol{D}(\boldsymbol{X}^\top \boldsymbol{X})^{-1})^{-1}\hat{\boldsymbol{\beta}}$ is an approximation for $\tilde{\boldsymbol{\beta}}$, we have the advantage of $\boldsymbol{D}$ being independent of $\boldsymbol{y}$. Hence, we can use all the results that we derived in section 3.3.1 where $\boldsymbol{B}$ was independent of $\boldsymbol{y}$.

3.4 Orthonormal Design Case

Now, we consider a simplified setting with an orthonormal design matrix $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$. Even if the orthonormal design matrix is unrealistic in practice, it helps us to understand the behavior of the suggested method compared to other shrinkage methods such as the lasso, ridge regression, and subset selection. Here, we follow similar steps as in Breiman (1995).

Since $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$ and least squares estimators of $\boldsymbol{\beta}$ are given by $\boldsymbol{X}^\top \boldsymbol{Y}$, we can simplify the generalized quadratic garrote coefficients as

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{I} + \lambda \boldsymbol{B})^{-1}\hat{\boldsymbol{\beta}},$$

with $\boldsymbol{B} = \text{diag}(d_j^2)$. Now the j^{th} estimated coefficient can be written as

$$\tilde{\beta}_j = \left(\frac{1}{1 + \lambda d_j^2}\right)\hat{\beta}_j. \tag{3.28}$$

Here, $1/(1+\lambda d_j^2)$ is the shrinkage factor for generalized quadratic garrote. One can obtain the corresponding quadratic garrote coefficients by substituting $d_j^2 = 1/\hat{\beta}_j$ in (3.28). Recall that $\hat{\beta}_j^2$'s are the OLS estimators, hence

$$\hat{\beta}_j^{(QG)} = \left(\frac{\hat{\beta}_j^{\,2}}{\hat{\beta}_j^{\,2} + \lambda} \right) \hat{\beta}_j, \tag{3.29}$$

where, $\hat{\beta}_j^{\,2}/(\hat{\beta}_j^{\,2} + \lambda)$ is the shrinkage factor for the quadratic garrote. Similarly, substituting $d_j^2 = 1$ for all d_j's in (3.28), we have shrinkage factor for the ridge estimators as $1/(1+\lambda)$.

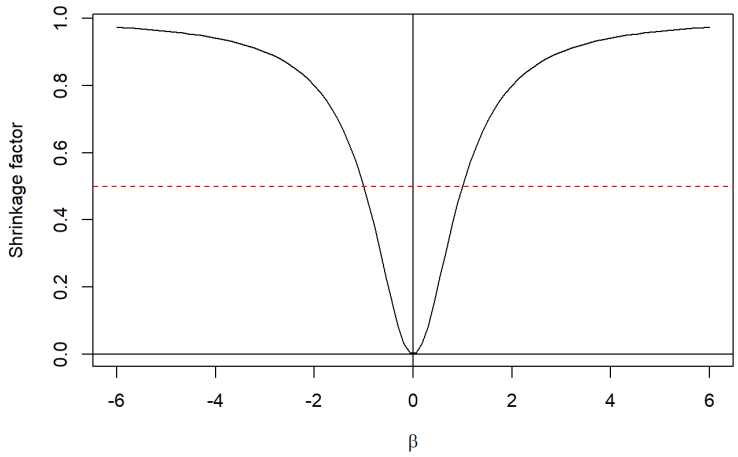


Figure 3.5: Shrinkage factor for the quadratic garrote with an orthogonal design matrix, when $\lambda = 1$. The dotted line represents the shrinkage factor of the ridge regression with $\lambda = 1$.

Figure 3.5 shows the nature of the shrinkage factor of the quadratic garrote against the value of the coefficient when $\lambda = 1$. The horizontal dotted line represents the corresponding shrinkage factor for the ridge regression. Ridge regression does a constant shrinkage for any value of β. Unlike ridge, quadratic garrote constraint applies more shrinkage for small values of β while it applies little shrinkage for larger coefficients. This might be favorable where we do not want to shrink larger coefficients but shrink the smaller coefficients more.

Now, assume the orthonormal design matrix, and consider y_i's which were generated with the linear model

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i,$$

with $\epsilon_i \sim$ iid $N(0, \sigma^2)$. It can be easily shown that the OLS estimator $\hat{\beta}_j$ can be written as

$$\hat{\beta}_j = \beta_j + Z_j,$$

where $Z_j \sim$ iid $N(0, \sigma^2)$. Let $\hat{\boldsymbol{\beta}}'$ be any estimator of $\boldsymbol{\beta}$. Following Breiman (1995), model error (ME) can be defined as

$$\text{ME}(\hat{\boldsymbol{\beta}}') = \frac{1}{p}\sum_{j=1}^{p}(\beta_j - \hat{\beta}'_j)^2, \tag{3.30}$$

where $\hat{\beta}_j{}'$ is the j^{th} element of $\hat{\boldsymbol{\beta}}'$. Let $\hat{\beta}'_j = c(\hat{\beta}_j, \lambda)\hat{\beta}_j$, where $c(.,.)$ is the shrinkage factor which is a function of $\hat{\beta}_j$ and λ. Hence, we have

$$\text{ME}(\lambda) = \frac{1}{p}\sum_{j=1}^{p}(\beta_j - c(\hat{\beta}_j, \lambda)\hat{\beta}_j)^2.$$

Using the fact that $\hat{\beta}_j = \beta_j + Z_j$

$$\text{ME}(\lambda) = \frac{1}{p}\sum_{j=1}^{p}(\beta_j - c(\beta_j + Z_j, \lambda)(\beta_j + Z_j))^2.$$

Assuming that p is large and β_j's are iid from the distribution $\mathcal{P}(d\beta)$, we can approximate the model error by

$$\text{ME}(\tilde{\boldsymbol{\beta}}) \approx E(\beta - c(\beta + Z, \lambda)(\beta + Z))^2.$$

Define the minimum model error (ME*) as

$$\text{ME}^* = \min_{\lambda}\text{ME}(\lambda).$$

Then the minimum model error is

$$\text{ME}^* \approx \min_{\lambda} E(\beta - c(\beta + Z, \lambda)(\beta + Z))^2.$$

For the generalized quadratic garrote, let d''s to be independent of $\hat{\boldsymbol{\beta}}$ and assume that we arbitrarily assign d' for each $\hat{\beta}_j$. We can show that

$$\text{ME}^* = \frac{E(\beta^2)}{E(\beta^2) + 1}. \tag{3.31}$$

To see this, consider the general QG estimator where d''s are arbitrarily assigned. Then the model error is given by

$$\begin{aligned} \text{ME} &= E\left(\beta - \tilde{\beta}\right)^2, \\ &= E\left(\beta - \frac{\beta + Z}{1 + \lambda d'^2}\right)^2, \end{aligned} \tag{3.32}$$

where, Z has a $N(0,1)$ distribution. Under regularity conditions, taking the first derivative of ME with respect to λ, and after some simplifications we have

$$\frac{\partial \text{ME}}{\partial \lambda} = \frac{2d'^2}{(1+\lambda d'^2)^2} E\left\{(\beta + Z)\beta - \frac{(\beta + Z)^2}{1 + \lambda d'^2}\right\}.$$

Since ME is convex, there exists a unique minimum for ME, ME* at λ^* such that,

$$\left.\frac{\partial \text{ME}}{\partial \lambda}\right|_{\lambda=\lambda^*} = 0.$$

Since $2d'^2/(1+\lambda d'^2)^2 \neq 0$, solving at λ^* we have

$$E\left\{(\beta + Z)\beta - \frac{(\beta + Z)^2}{1 + \lambda^* d'^2}\right\} = 0.$$

By solving for λ^*, we have

$$\lambda^* = \frac{1}{d'^2 E(\beta^2)}.$$

Substituting λ^* in (3.32) and after further simplification, we obtain

$$\text{ME}^* = \frac{E(\beta^2)}{E(\beta^2) + 1}. \tag{3.33}$$

Since (3.33) is independent of d', this result is directly valid for ridge regression. However, when d''s are not independent of $\hat{\boldsymbol{\beta}}$, it is difficult to obtain a closed form for ME*. Instead, we obtain an approximate solution for ME* through simulation.

To investigate ME*, consider the family of distributions $\mathcal{P}(d\beta) = \theta\delta(d\beta) + (1-\theta)Q(d\beta, \sigma^2)$, where $\delta(d\beta)$ is a mass concentrated at zero and $Q(d\beta, \sigma^2) \sim N(0, \sigma^2)$. $\theta \in [0,1]$ and $\sigma \in [0,5]$. Breiman (1995) used the same family of mixture distributions to investigate the properties of the non-negative garrote. This family is suitable for our study as well since it makes a good playing ground to compare the performance of quadric garrote estimators with other methods under few different scenarios.

First, we fixed σ and simulated 10000 β values from $d(\beta)$ at each λ on a grid of values ranged from 0 to 10 by 0.01. Then, at each λ, ME was evaluated and then, ME* was selected as the minimum ME. This

was repeated for a set of $\sigma \in [0, 5]$. Entire process was repeated for each shrinkage method and the results are presented in Figure 3.6.

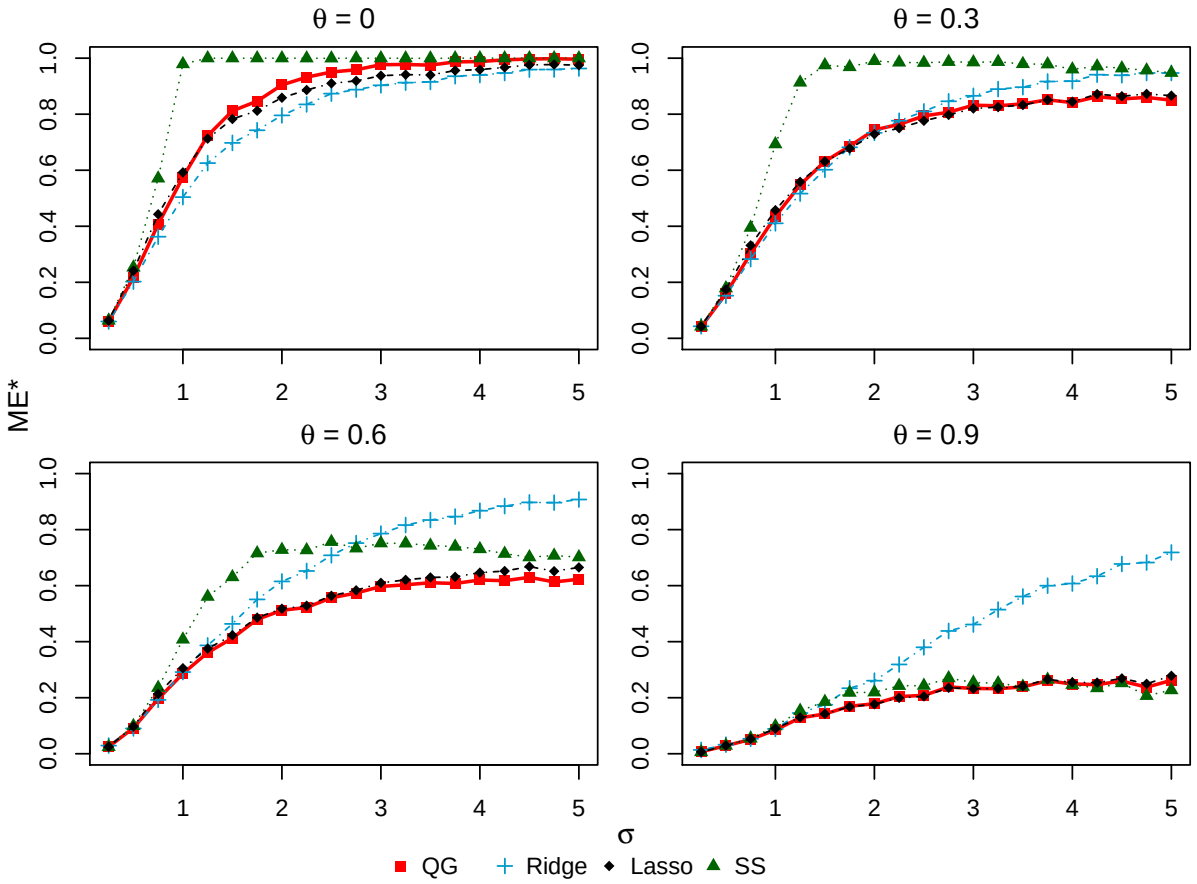


Figure 3.6: Plot of ME* vs σ for different shrinkage methods under the orthonormal design assumption.

When $\theta = 0$, the proportion of zero coefficients in the model are almost zero. In this case, we see that all the other methods perform better than the subset selection for small σ values. However, as σ increases, all the methods tend to give similar ME* values. Ridge regression performs better in this scenario than the lasso and the quadratic garrote method.

In the second case, when there are 30% of zero coefficients in the model, ridge regression performs poorly as σ increases. Quadratic garrote method and the lasso give similar ME* values. In the last plot in Figure 3.6, where there are many zero coefficients in the true model, ME* values for the ridge regression models increase dramatically while all the other methods give similar results. As a whole, quadratic garrote estimator performs uniformly well compared to ridge regression method. In this

section, we only considered the orthogonal design where everything is simple. In the proceeding sections, we further study the properties of these methods for the general setting through simulation.

3.5 Simulation Study

In this section, we perform a simulation study considering different settings which resemble real-life problems such as multicollinearity issue, high dimensional setting, etc. The performance of quadratic garrote will be compared with ridge and lasso estimation methods in each setting. OLS estimation will be used as the benchmark. Consider the following three settings:

1. Sparse setting,
2. Nearly-sparse setting,
3. High dimensional setting.

In each setting, data were generated using the model $y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$, $i = 1, 2, ..., n$. Note that we can write the linear regression model as $y_i = \boldsymbol{x}_i^T\boldsymbol{\beta} + \epsilon_i$. $\boldsymbol{x}_i$'s were generated from a $MVN(\mathbf{0}, \Sigma)$, where ij^{th} entry of Σ is $\rho^{|i-j|}$ and $\rho = 0.5$. The sample size is 100 in each setting. 10-fold cross-validation method was used to select the best tuning parameter for each of the shrinkage method. Finally, mean-squared prediction error was used to compare the prediction accuracy between models with 5-fold cross-validation.

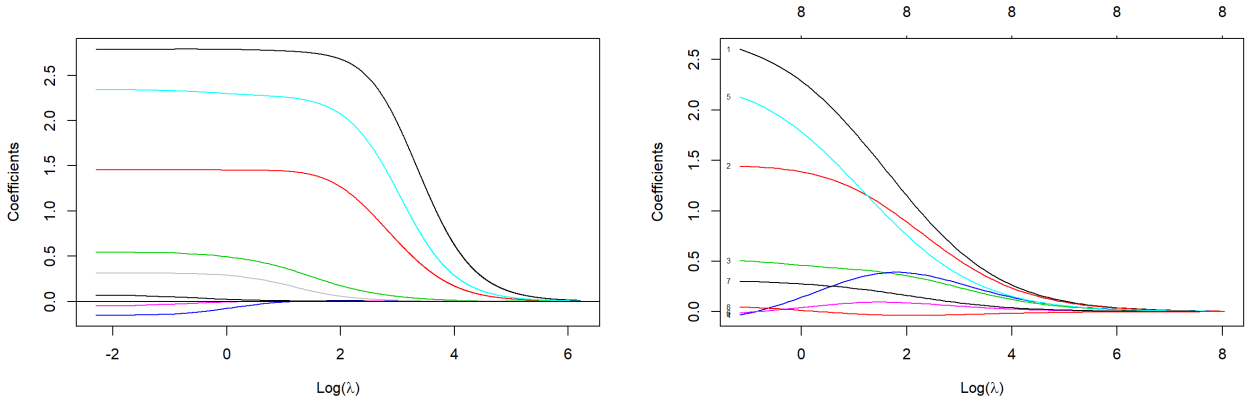
3.5.1 Sparse Setting

This setting has been used in Tibshirani (1996) to compare the performance of the lasso with other shrinkage methods. Our intention is to see how the quadratic garrote model performs when there is a considerable fraction of zeros in$\boldsymbol{\beta}$. We set the true population coefficient vector to be $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^\top$. The random error was set to have $\sigma = 3$. The signal to noise ratio of the sample data was approximately 2. Consider an additive model $Y = f(X) + \epsilon$. Then, according to Hastie et al. (2009) the signal to noise ratio is defined as,

$$\text{Signal to noise ratio} = \frac{Var(f(X))}{Var(\epsilon)}. \tag{3.34}$$

The signal to noise ratio for the multiple linear model is given by $Var(X^\top\beta)/Var(\epsilon)$. The above given value of the signal to noise ratio of 2, is the estimated signal to noise ratio of the sample data, calculated as the ratio of sample variance of $\boldsymbol{X\beta}$ vector to the variance of the residual error vector.

Figure 3.7 shows the trace plots of quadratic garrote estimates and ridge estimates. We can see an interesting behavior of quadratic garrote estimators (Figure 3.7a) compared to ridge estimators (Figure 3.7b). In the quadratic garrote trace plot, we notice that the larger coefficients do not shrink until the tuning parameter (λ) is very large. However, the smaller coefficients approach zero very fast even for a very small λ with almost no effect on large coefficients. This property cannot be seen with the ridge regression. The ridge regression method shrinks all coefficients from the beginning.



(a) (b)

Figure 3.7: Trace plot of quadratic garrote estimators (a) and ridge estimators (b) for the sparse setting.

Figure 3.8 shows the mean squared error for quadratic garrote against λ. The best coefficients were selected to have minimum MSE (Vertical line on Figure 3.8 represents the location minimum MSE). The estimates are given in the table 3.1. Quadratic garrote estimates for the true parameters 3 and 2 are very close to their OLS estimates than the ridge estimates. On the other hand, the estimated coefficients for zero valued parameters are very small in the quadratic garrote than the ridge estimates.

Table 3.2 summarizes the mean squared errors of each of our model. For the sparse setting, quadratic garrote gives the least prediction errors than all the other methods. It even has better prediction accuracy than the lasso, which has the second best MSE.

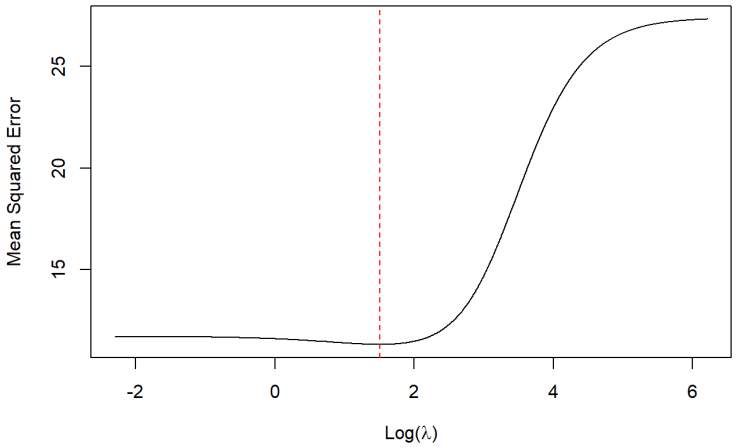


Figure 3.8: Mean squared error for quadratic garrote under the sparse setting.

Table 3.1: Estimated coefficients under the sparse setting.

	β_j	OLS	QG	Ridge	Lasso
X1	3.00	2.7865	2.7456	2.5871	2.6667
X2	1.50	1.4571	1.3995	1.4421	1.4136
X3	0.00	0.5471	0.2717	0.5042	0.3809
X4	0.00	-0.1553	0.0038	-0.0231	0
X5	2.00	2.3489	2.2097	2.1141	2.1432
X6	0.00	-0.0522	0.0008	-0.0075	0
X7	0.00	0.3161	0.1126	0.3000	0.2130
X8	0.00	0.0707	0.0030	0.0449	0

Table 3.2: Mean squared errors of the models under sparse setting.

	MSE
OLS	12.2485
Quadratic garrote	11.7609
Ridge	12.1020
Lasso	11.7974

3.5.2 Nearly-sparse Setting

In this setting, we set the population coefficients vector to have a few large values and others are set to have values closer to zero. Let $\boldsymbol{\beta} = (3, 1.5, z, z, 2, z, z, z, z, z)^\top$, where $z \sim Unif(0, b)$, $0 < b \leq 1$. We can set z coefficients to have smaller values by selecting a very small b. The setting was repeated for $b \in \{0.1, 0.5, 1\}$, and the random error σ was selected to be $\sqrt{2}$ such that the signal to noise ratio is around 10 for each case. A similar setting has been suggested in Zhang et al. (2014).

Table 3.3: Estimated QG coefficients and prediction errors under the nearly-sparse setting

(a) $b = 1$

	Beta	OLS	QNNG	Ridge	Lasso
X1	3.0000	2.9778	2.9776	2.8094	2.9419
X2	1.5000	1.7939	1.7751	1.7103	1.7455
X3	0.1966	-0.0158	-0.0001	0.0879	0
X4	0.7164	0.8819	0.8559	0.8474	0.8375
X5	2.0000	2.1116	2.1510	1.9420	2.1165
X6	0.3621	0.3048	0.2457	0.4195	0.2757
X7	0.3911	0.6722	0.6691	0.6455	0.6427
X8	0.8133	0.6658	0.6467	0.6483	0.6372
X9	0.4280	0.5192	0.5016	0.5416	0.5003
X10	0.9592	0.8133	0.8130	0.7347	0.7780

(b) $b = 0.5$

	Beta	OLS	QNNG	Ridge	Lasso
X1	3.0000	2.8638	2.8721	2.6576	2.8328
X2	1.5000	1.6368	1.6167	1.5636	1.6081
X3	0.1539	0.1109	0.0162	0.1905	0.0537
X4	0.1288	-0.1171	-0.0019	0.0785	0
X5	2.0000	2.0214	2.0270	1.7472	1.9712
X6	0.2762	0.4946	0.4278	0.5760	0.4568
X7	0.0282	-0.0369	0.0024	0.0062	0
X8	0.2343	0.1597	0.0629	0.1342	0.1467
X9	0.2419	0.0654	0.0081	0.0188	0.0044
X10	0.4062	0.5987	0.5916	0.5993	0.5940

(c) $b = 0.1$

	Beta	OLS	QNNG	Ridge	Lasso
X1	3.0000	2.9236	2.9779	2.6933	2.8817
X2	1.5000	1.3315	1.2467	1.3001	1.2416
X3	0.0336	0.2110	0.0464	0.2706	0.0631
X4	0.0464	-0.2883	-0.0655	-0.1734	0
X5	2.0000	2.3813	2.2207	2.0770	2.0929
X6	0.0061	-0.0905	0.0000	0.0034	0
X7	0.0197	0.0183	0.0001	0.0442	0
X8	0.0474	0.1739	0.0228	0.1394	0
X9	0.0301	-0.2275	-0.0123	-0.1473	0
X10	0.0607	0.1711	0.0346	0.1691	0.0521

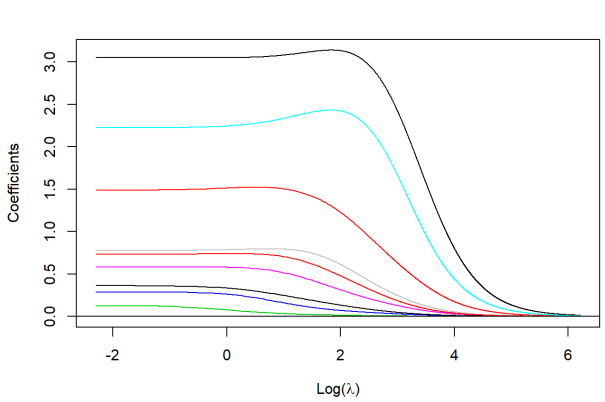
The trace plots for the quadratic garrote coefficient estimates and the ridge estimates for different b values are shown in Figure 3.9. The behavior of trace plots for both quadratic garrote and ridge are similar to their behavior under the sparse setting. In all values of b, we see that the quadratic garrote does not shrink the larger coefficients unless λ is very large. On the other hand, smaller coefficients shrink towards zero very fast even for a small λ.

Estimated quadratic garrote coefficients at the best λ which was determined by cross validation for each scenario, are presented in Table 3.3. When $b = 1$, we do not observe much shrinkage on each coefficient in any shrinkage method except for smaller coefficients. However, when $b = 0.5$ and $b = 0.1$, the coefficients are shrunk in all the models. In these scenarios, unlike ridge regression, the quadratic garrote shrinks the smaller coefficient almost to zero with no shrinkage on the larger coefficients.

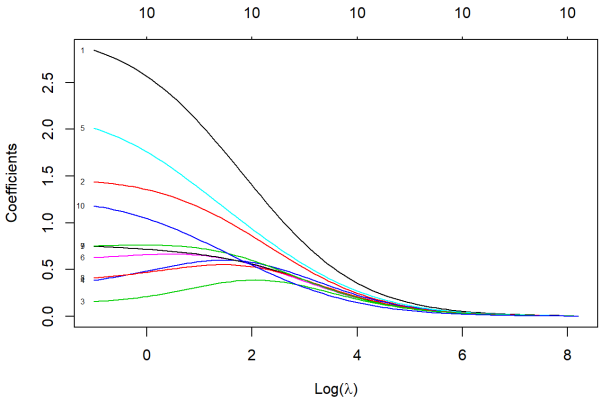
Mean squared cross-validation errors for each approach is summarized in Table 3.4. We see that, for all values of b that we used, the quadratic garrote performs better than the ridge regression approach. As b becomes smaller, quadratic garrote does better prediction than all the other methods. In all the cases, we see that the ridge regression approach gives a highest prediction errors. As b becomes smaller, prediction errors of ridge regression approach is even higher than the OLS approach.

Table 3.4: Mean squared errors of the models under nearly sparse setting.

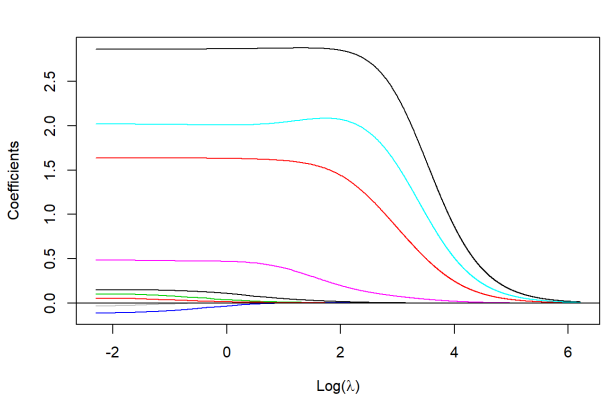
	MSE		
	$b = 1$	$b = 0.5$	$b = 0.1$
OLS	3.0830	2.3372	2.1892
Quadratic garrote	3.0643	2.2154	2.1114
Ridge	3.0753	2.3774	2.2375
Lasso	3.0330	2.2203	2.1419



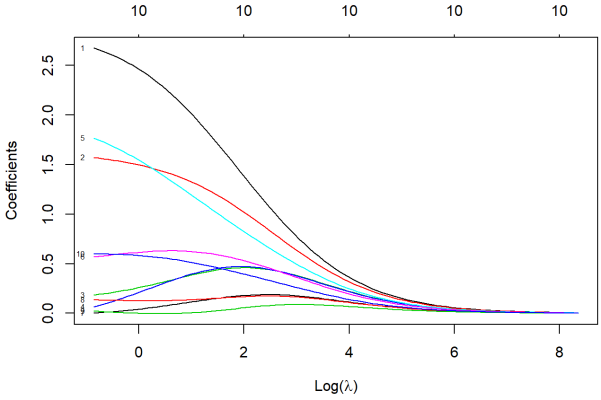
(a) $b = 1$



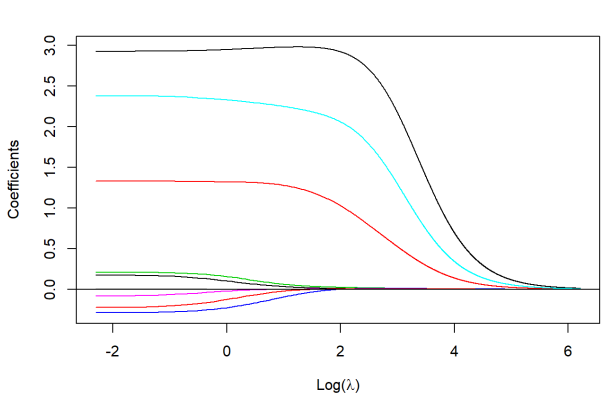
(b) $b = 1$



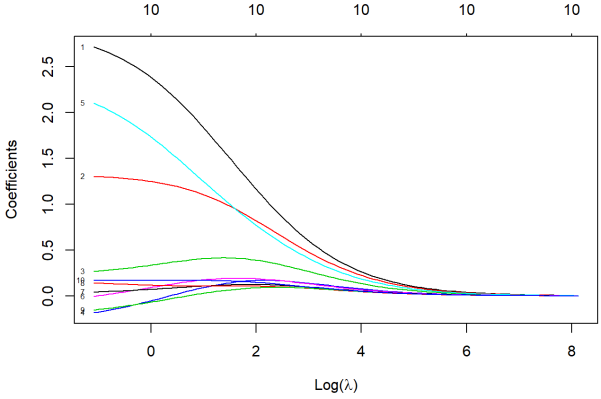
(c) $b = 0.5$



(d) $b = 0.5$



(e) $b = 0.1$



(f) $b = 0.1$

Figure 3.9: Trace plot of quadratic garrote estimates (Left) and ridge estimates (Right) of the nearly sparse setting.

3.5.3 High Dimensional Setting

As we discussed in Chapter 1, the problems associates with OLS estimator in high dimensional setting are more serious than they are in the lower dimensions. Hence, it is important to examine how the suggested model performs in the high dimensional setting. To this ens we consider three scenarios.

In the first scenario, the coefficient vector consists of the majority of very small coefficients and some relatively large coefficients which represent the important predictors. Let's take

$$\boldsymbol{\beta} = (z, z, ..., z, 1, 1, 1, 1, 1, z, z, ..., z, 4, 4, 4, 4, 4)^\top,$$

where z's are small positive coefficients independently generated from a $Gamma(1, 10)$ distribution. There are 30 parameters in four blocks, where first and third blocks contain 10 z values in each block. σ was set to be 5, which gave approximately a signal to noise ratio of 8 in the sample.

In the second scenario, we add some more moderately large coefficients which have the same number of smaller coefficients. Here we consider

$$\boldsymbol{\beta} = (3, 3, 3, 3, 3, z, z, ..., z, 1, 1, ..., 1, z, z, ..., z, 2, 2, 2, 2, 2)^\top,$$

where z's are iid from a $Gamma(1, 10)$ distribution. There are 40 parameters in five blocks, where the second to fourth blocks contain 10 values in each block. σ was set to be 5, and the signal noise ratio was approximately 11.

In the third scenario, coefficient vector contains many smaller coefficients, and some moderately large coefficients with few very large coefficients. We set

$$\boldsymbol{\beta} = (z, z, ..., z, 5, 5, 5, z, z, ..., z, 1, 1, 1, 1, 1, z, z, ..., z, 10, 10)^\top,$$

where z's are iid from a $Gamma(1, 10)$ distribution. It contain 40 parameters in six blocks, where each of the z block contain 10 values. $\sigma = 7$ and the signal noise ratio was approximately 10.

The trace plot of each scenario is shown in Figure 3.10. The dotted vertical line represents the λ with respect to the model with minimum MSE obtained using 10-fold cross validation. Even if the quadratic garrote does not do subset selection, when we observe the trace plots of quadratic garrote estimates (Figure 3.10 (a)(c)(e)), it can be seen that at the best λ, a large number of coefficients has been shrunken towards zero with a very little or almost no effect on the large coefficients. This is an important property in the high dimensional setting because we can have a better prediction while keeping the most important variables unchanged. But in the ridge trace plots (Figure 3.10 (b)(d)(f)), we can see that there all the coefficients shrink towards zero similarly.

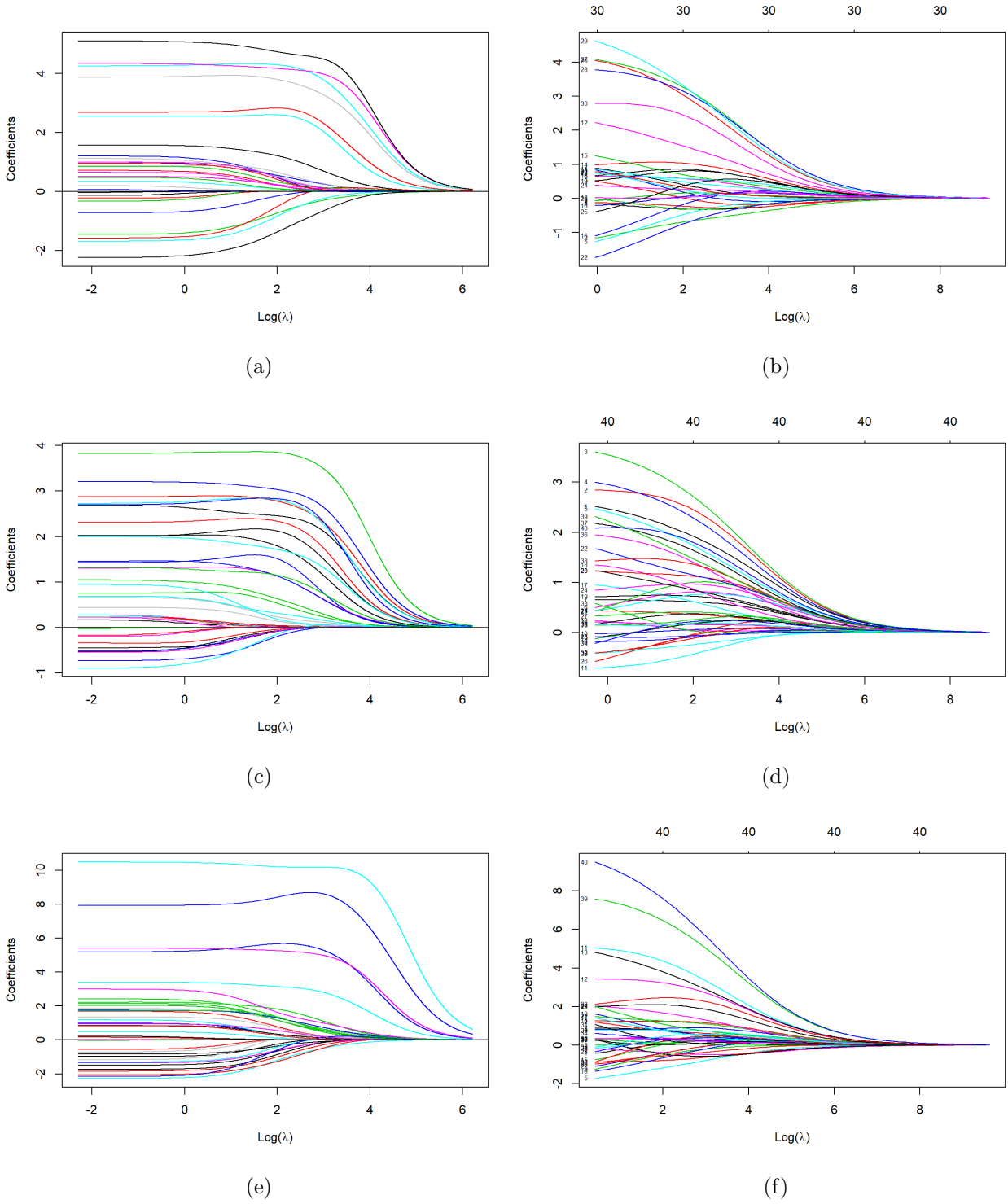


Figure 3.10: Trace plot of quadratic garrote estimates (Left) and ridge estimates (Right) of the high dimensional setting where (a)(b) - Scenario 1, (c)(d) - Scenario 2, (e)(f) - Scenario 3 (Dotted line indicates the best lambda w.r.t. minimum MSE).

The prediction error of each model is presented in the Table 3.5. In the first scenario, where there are many small coefficients with few moderately large coefficients, the ridge regression does slightly better than all the other approaches. But in Scenario 2, where there are the same number of small coefficients and moderately large coefficients, quadratic garrote performs slightly better than the ridge. However, in Scenario 3, where there are a large number of small coefficients with a few large coefficients and few very large coefficients, the quadratic garrote performs really well with a significantly low prediction error compared to the ridge regression model. In this case, quadratic garrote model is even competitive with the lasso. However, the lasso outperforms both of the quadratic garrote and the ridge regression specially in the high dimensional setting in terms of the prediction error.

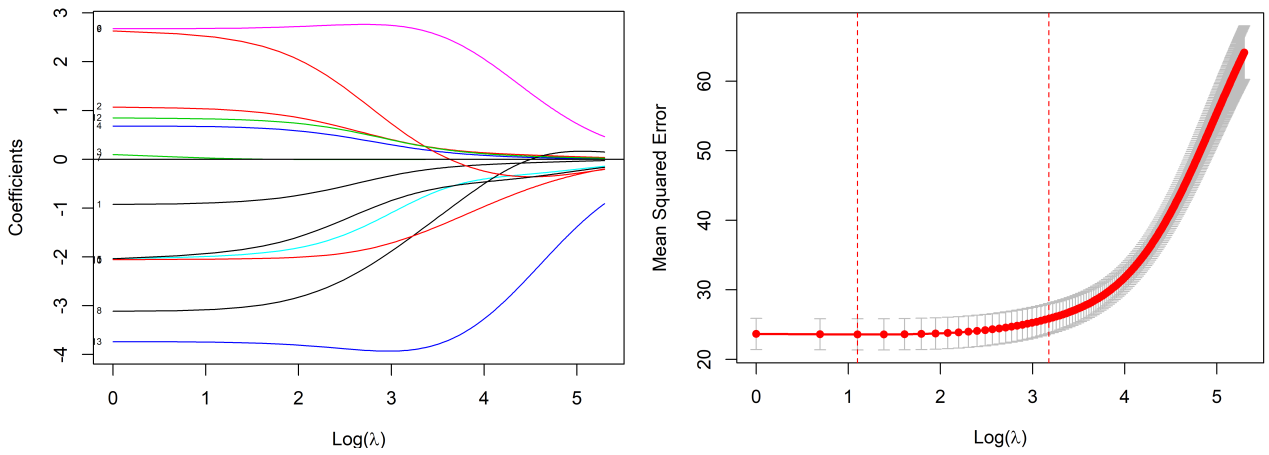
Table 3.5: Mean squared errors of the models under high dimensional setting.

	MSE		
	Scenario 1	Scenario 2	Scenario 3
OLS	33.8628	35.5502	101.8177
Quadratic garrote	32.5943	25.5129	70.9068
Ridge	29.6387	25.8657	84.4298
Lasso	30.6148	21.5234	68.9327

3.6 Example 1: The Boston Housing Dataset (Continued)

In this section, we further illustrate the use of the generalized garrote and the quadratic garrote with the Boston housing dataset that we described in Chapter 2. Here as well suppose that we want to build a model to predict medv with all the other variables as predictors. The solution path for the quadratic garrote can be found in Figure 3.11a. In the ridge regression solution path in Figure 2.13a, we noticed that, as we increase λ, all the coefficients shrink towards zero at a similar rate from the beginning. On the other hand, quadratic garrote shrinks the smaller coefficient faster than larger coefficients. Largest coefficients shrinks only for large values of λ and resist to shrink otherwise. The 10-fold cross validation error plot of quadratic garrote for Boston dataset can be found in Figure 3.11b. Two vertical dotted lines represent the λ values with respect to the minimum prediction error and prediction error with the one standard error rule. The error bars represent the standard deviation of the mean squared error values at each λ.

Suppose the researcher who conducts the study thinks that the crime rate, distance to work and number of rooms in the dwelling are the key factors which determine the house price and he does not want to shrink the effect of those variables in the model. Also, he wants only a moderate shrinkage on



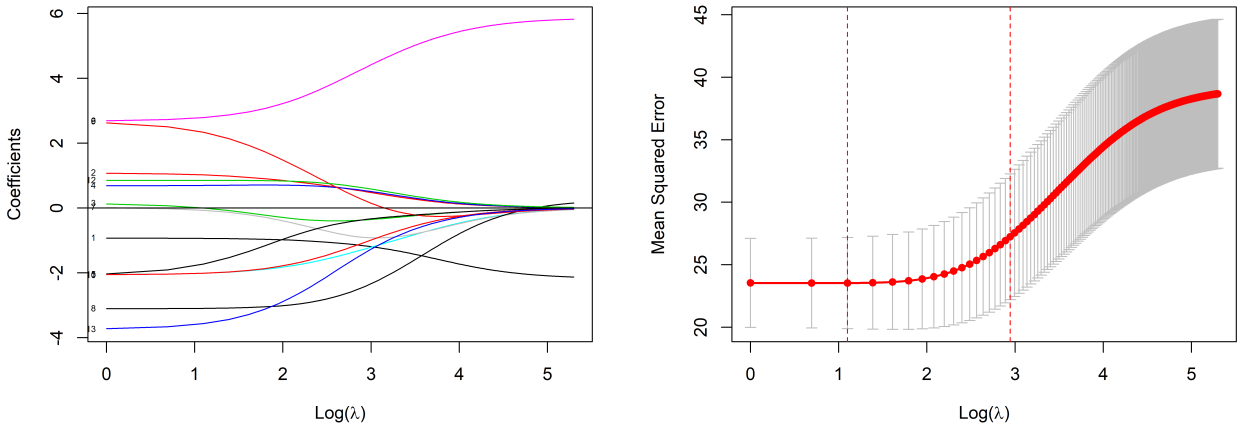
(a) (b)

Figure 3.11: (a) Solution path of quadratic garrote for Boston dataset. (b) Cross validation error of quadratic garrote for Boston dataset.

the variables which represent air pollution, the age of the dwelling and accessibility to highways. For this situation, the researcher can use the generalized quadratic garrote with a user-defined vector of shrinking factors, such as $\{d_j\} = (0, 1, 1, 1, 0.5, 0, 0.5, 0, 0.5, 1, 1, 1, 1)^\top$. The shrinking factor $d_j = 0$ avoids imposing any penalty on the corresponding coefficient estimate. Higher the value of d_j, higher the shrinkage on the corresponding coefficient. The solution path of the generalized quadratic garrote with the above defined shrinking factors can be found in Figure 3.12a and the prediction error is presented in Figure 3.12b. We observe that as we increase λ, some model coefficients actually increase instead of shrinking towards zero. Those are the parameters which we omit from shrinking by setting the corresponding d_j's to be exactly zero. One can confirm those variables which do not shrink by observing the estimated model coefficients in Table 3.6.

Table 3.6 summarizes the results of two quadratic garrote models along with OLS, the ridge and the lasso results. All models have been evaluated at the λ which gives the minimum prediction error. Compared to the ridge regression and the lasso, the quadratic garrote does minimum shrinkage on the larger coefficients and on the other hand, it shrinks smaller coefficients by a larger factor than the ridge or the lasso. Generalized quadratic garrote, which we arbitrarily defined shrinking factors, does what we intended. It does not shrink the coefficients of the variables crim, rm, and dis. Those are the variables corresponding to the d_j's that we set to be zero. It applies minimum shrinkage on the coefficients of nox, age, and rad since we set a smaller shrinking factor $d_j = 0.5$ for them. However, other coefficients where we set a larger d_j values has been shrunk towards zero by a larger factor compared to other shrinkage methods. One of the most important thing that we notice is, that the prediction errors of

both of the suggested approaches are slightly smaller than the ridge regression.



(a) (b)

Figure 3.12: (a) Solution path of generalized quadratic garrote for Boston dataset. (b) Cross validation error of generalized quadratic garrote for Boston dataset.

Table 3.6: Coefficient estimates and cross validation errors for each shrinkage method.

	OLS	QG	GQG (d_j^2)	Ridge	Lasso
crim	-0.9291	-0.8927	-0.9373 (0.00)	-0.7441	-0.8600
zn	1.0826	1.0272	1.0205 (1.00)	0.7450	0.9834
indus	0.1410	0.0221	0.0008 (1.00)	-0.2765	0.0000
chas	0.6824	0.6688	0.6960 (1.00)	0.7381	0.6836
nox	-2.0588	-1.9845	-2.0121 (0.25)	-1.3396	-1.9101
rm	2.6769	2.6836	2.7907 (0.00)	2.8215	2.7096
age	0.0195	0.0001	-0.0718 (0.25)	-0.1113	0.0000
dis	-3.1071	-3.0787	-3.0971 (0.00)	-2.3029	-2.9734
rad	2.6649	2.5045	2.3369 (0.25)	1.2771	2.2667
tax	-2.0788	-1.9181	-1.7312 (1.00)	-0.9274	-1.7098
ptratio	-2.0626	-2.0454	-2.0110 (1.00)	-1.8367	-2.0193
black	0.8501	0.8277	0.8506 (1.00)	0.8258	0.8281
lstat	-3.7473	-3.7486	-3.5660 (1.00)	-3.3445	-3.7313
MSE	23.1516	23.0843	23.1137	23.5036	23.1050

In Section 3.5, we saw that the quadratic garrote uniformly performs well under different simulation settings. In this section, we further confirmed that the QG method and the generalized QG approach also performed well with real data. The prediction accuracy of suggested methods was better than the ridge regression for most of the cases and, they were competitive to lasso as well. Our next chapter will discuss another interesting substitute for the ridge regression.

Chapter 4

LINEX Regression

In this chapter, we first explore the possibility of using different loss functions as the penalty in the non-negative garrote. We introduce the LINEX regression method as a novel shrinkage method, which will further broaden the scope of shrinkage methods. Furthermore, we study the performance of the suggested approach with a simulation study under different settings and the real data application with the Boston Housing dataset.

4.1 Introduction

In the previous chapters, we studied different types of shrinkage methods and their impact on the shrinkage of the coefficients towards zero. As we learned, the nature of the shrinkage on coefficients depends on how we define the restriction on the space of $\boldsymbol{\beta}$ or, in other words, the penalty term that we add to sum of the squared errors when estimate coefficients. Hence, before presenting the LINEX regression approach, as a motivation, we will explore some of the popular loss functions and their usefulness as the penalty in the non-negative garrote. Consider the minimization problem,

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} f(c_j), \tag{4.1}$$

where $\hat{\beta}_j$ is the j^{th} OLS estimate and $f(c_j)$ is a known function of c_j. Here, $\lambda \sum_{j=1}^{p} f(c_j)$ is called the penalty term, and $c_j \hat{\beta}_j$ is the new estimates of β_j. The penalty function, $f(c_j)$ plays an important role here. We cannot just use any function. To understand the role of this penalty, consider the following $f(c_j)$ functions.

1. No-negative garrote: $f(c_j) = c_j, c_j > 0$,

2. Quadratic garrote: $f(c_j) = c_j^2$,

3. Chi-square: $f(c_j) = (c_j - 1)^2/c_j$,

4. Exponential: $f(c_j, \alpha, \lambda) = \frac{\lambda^2}{\alpha}\left\{1 - \exp\left(-\frac{\alpha c_j}{\lambda}\right)\right\}$, where $\alpha > 0$ is a constant,

5. LINEX: $f(c_j, \alpha) = e^{\alpha c_j} - \alpha c_j - 1$, where $\alpha > 0$ is a constant,

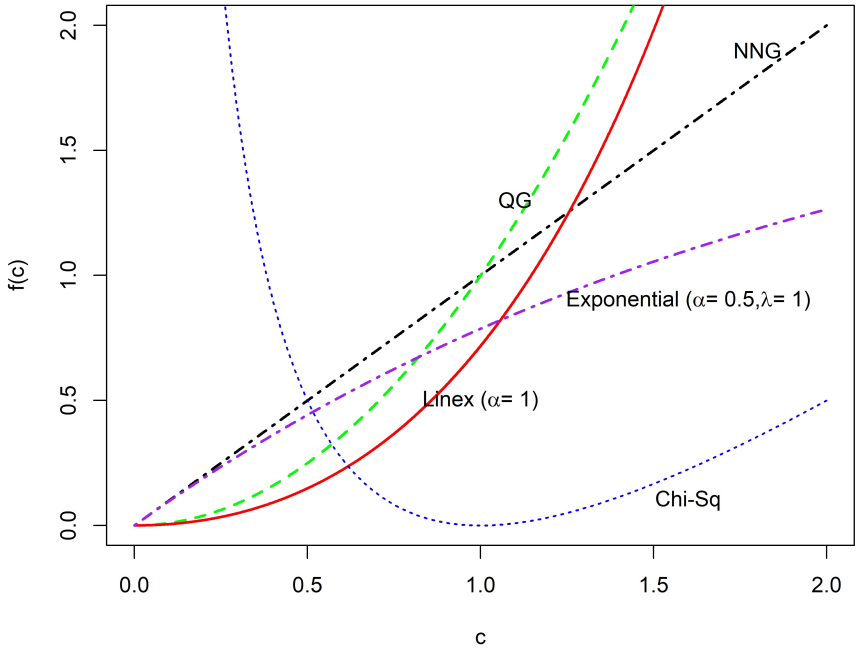


Figure 4.1: Different penalty functions.

These functions can be found in Figure 4.1. In the quadratic garrote estimation, we minimized the quadratic loss function, SSE, under a quadratic constraint or equivalently, we minimized

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} c_j\hat{\beta}_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p} c_j^2.$$

Recall that $c_j\hat{\beta}_j = \tilde{\beta}_j$, is the new estimates of β_j. The quadratic penalty will encourage $\tilde{\beta}_j$ to have smaller values, by increasing the penalty on SSE for larger values of $\tilde{\beta}_j$ in a quadratic manner. Hence, large $\tilde{\beta}_j$'s will have very large penalties compared to the non-negative garrote penalty. Even the least squares estimates, that is when $c_j = 1$ will result in a penalty on the loss function. Hence, this penalty function leads the estimates to have smaller values than OLS estimates. However, the penalty function rapidly drops as c_j decreases from one and then it slowly approaches zero as c_j approaches zero. However,

since the penalty is very small for small c_j values which are close to zero, coefficient estimates will not be forced to be exactly zero.

Chi-square penalty, on the other hand, acts in an inverse way. This is an asymmetric function around one. If we perform regularize regression with this penalty, we will have coefficient estimates which are not far away from OLS estimates. Chi-square function gives a huge penalty for c_j's closer to zero and, also gives a large penalty as coefficients become larger than OLS estimates. This function does not allow coefficient to shrink towards zero at all. Hence, chi-square penalty is not appropriate for the shrinking purpose.

The exponential penalty function is different from the above two. This function is concave. Like the non-negative garrote penalty, the exponential penalty can be used for subset selection purpose because of the concave property. As we see in Figure 4.1, the exponential loss can set c_j's to be exactly zero. In the recent paper, Breheny (2015) used the exponential loss function as the penalty term on the regression coefficients with the quadratic loss as below

$$Q(\beta_0, \beta_1, ..., \beta_p) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \sum_{j=1}^{p} f(\beta_j, \alpha, \lambda). \tag{4.2}$$

where $f(\beta_j, \alpha, \lambda)$ is the exponential penalty function of β_j. Also Breheny (2015) use the exponential function as the outer function of the grouped lasso penalty function.

The focus of this chapter is the linear exponential (LINEX) penalty which is defined with the parameter, α. The function is first introduced in Varian (1975). Ohtani (1995) also discusses a feasible generalized ridge regression estimators with the LINEX loss function. Akdeniz (2004) uses the LINEX loss function as an asymmetric substitute for the squared error loss function. The nature of the function for $\alpha = 1$ can be seen in Figure 4.1. In this case, LINEX function gives a very large penalty for large values of c. Unlike other functions, we can change the shrinking rate in the LINEX regression by choosing different α values. As we see in Figure 4.2, for a large value of α, LINEX penalty can shrink the coefficient estimates more than quadratic garrote or non-negative garrote. On the other hand, for a small value of α, LINEX penalty applies a less amount of shrinkage.

4.2 LINEX Regression

Consider the minimizing the quadratic loss function; SSE, under the LINEX penalty. That is, we estimate $\{c_j\}_{j=1}^{p}$ to minimize

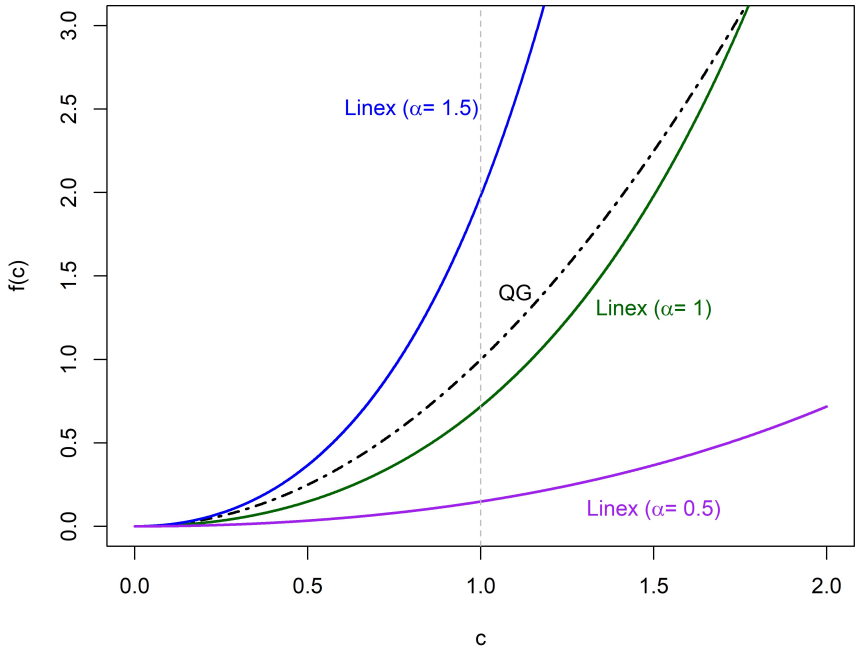


Figure 4.2: LINEX function for different α values. Dotted curve represent the QG function.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} c_j\hat{\beta}_j x_{ij})^2,$$

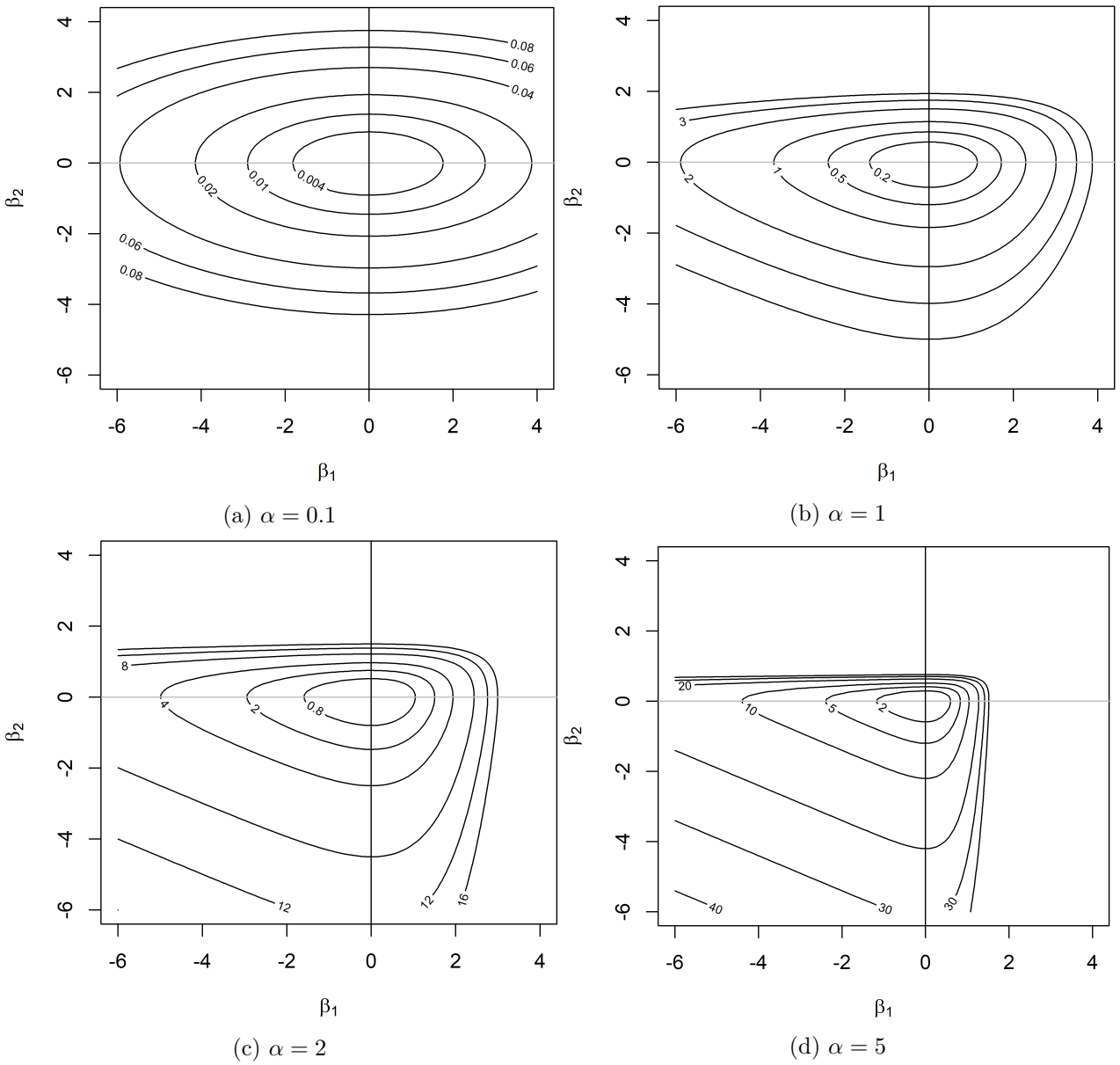
under the constraint $\sum_{j=1}^{p}(e^{\alpha c_j} - \alpha c_j - 1) \leq s$. This is equivalent to minimizing

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} c_j\hat{\beta}_j x_{ij})^2 + \lambda\sum_{j=1}^{p}(e^{\alpha c_j} - \alpha c_j - 1), \tag{4.3}$$

where, λ is selected such that $\sum_{j=1}^{p}(e^{\alpha c_j} - \alpha c_j - 1) = s$, and α is a positive constant. We will call the approach to be the LINEX regression. Then, ${\beta_j}^{Lx} = c_j\hat{\beta}_j$ are the new LINEX regression coefficients. Writing $c_j = {\beta_j}^{Lx}/\hat{\beta}_j$, ${\beta_j}^{Lx}$ is obtained as

$$\hat{\boldsymbol{\beta}}^{Lx} = \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left(e^{\alpha\frac{\beta_j}{\hat{\beta}_j}} - \alpha\frac{\beta_j}{\hat{\beta}_j} - 1\right)\right\}. \tag{4.4}$$

For the two predictor case, the nature of the LINEX penalty is illustrated in Figure 4.3 for $\hat{\boldsymbol{\beta}} = (2,1)^\top$ and different α values. As we see, LINEX penalty is asymmetric. Hence, it shrinks each OLS coefficient estimate differently. Similarly to other shrinkage methods, LINEX regression also restricts the parameter space by imposing the LINEX penalty on the parameter estimates. However, the shape of the LINEX



(a) $\alpha = 0.1$ (b) $\alpha = 1$

(c) $\alpha = 2$ (d) $\alpha = 5$

Figure 4.3: LINEX penalty for different s and α for a multiple regression model $y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$.

penalty changes with α. For small values of α, the penalty is roughly an ellipse and the shape distorts as we increase α. To further illustrate the idea, consider the same example data with two predictors X_1 and X_2, which we saw in Figure 2.2 in Section 2.2. Figure 4.4 shows the solution for the LINEX problem for the example dataset along with the LINEX penalty with $\alpha = 5$ at $s = 4$ and the contours of SSE.

The optimization problem (4.4) is convex. However, we cannot derive a closed form solution for $\hat{\boldsymbol{\beta}}^{Lx}$.

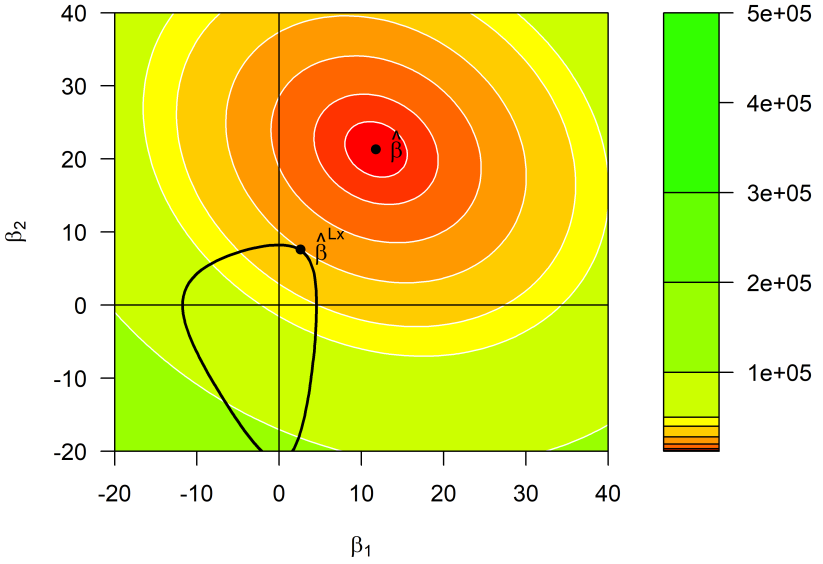


Figure 4.4: Contour plot of SSE with the LINEX penalty and the solution for $s = 3$.

Hence, we have to use a numerical optimization technique. We will use the 'optim' package in R with the 'L-BFGS-B' algorithm suggested by Byrd et al. (1995).

4.3 Variance and Bias of the LINEX Regression Estimator

Even if we are unable to of derive an exact solution for the LINEX regression estimator in a closed form, we can derive an approximate closed form solution. Consider the second order Taylor series approximation of $e^{\alpha c_j}$ is given by

$$e^{\alpha c_j} \approx 1 + \alpha c_j + \frac{(\alpha c_j)^2}{2}. \tag{4.5}$$

Then, using the approximation, we can rewrite (4.3) as

$$\hat{\boldsymbol{\beta}}^{Lx} \approx \underset{\beta_0,\beta_1,\ldots,\beta_p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \frac{(\alpha c_j)^2}{2} \right\}. \tag{4.6}$$

This is simply the quadratic garrote problem with the penalty term scaled by α^2. Then the approximate

closed form solution for the LINEX regression estimator is given by

$$\hat{\boldsymbol{\beta}}^{Lx}(\lambda,\alpha) \approx \left(\mathbf{X}^\top\mathbf{X} + \frac{\lambda\alpha^2}{2}\mathbf{B}\right)^{-1}\mathbf{X}^\top\mathbf{y}, \tag{4.7}$$

where $\boldsymbol{B} = \text{diag}(1/\hat{\beta}_j^{\,2})$ and $\hat{\beta}_j$'s are the OLS estimators. Hence, the variance and bias of LINEX regression estimator can be derived in the same way we did for the quadratic garrote estimator in Section 3.3. Since LINEX regression estimator is a function of α, we can see that, for a large α, the LINEX coefficient estimates will shrink towards zero faster than the quadratic.

4.4 Simulation Study

To evaluate the performance of the LINEX regression approach, we perform a simulation study under the same set of settings that we used in Section 3.5. Consider the following three settings:

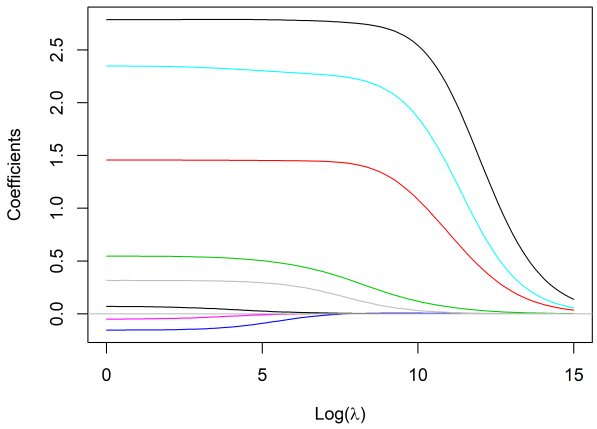
1. Sparse setting,
2. Nearly-sparse setting,
3. High dimensional setting.

The data for each setting was generated using the model $y_i = \boldsymbol{x}_i^T\boldsymbol{\beta}+\epsilon_i$, where $\epsilon_i \sim N(0,\sigma^2)$, $i = 1, 2, ..., n$. $\boldsymbol{x}_i$'s were generated from a $MVN(\mathbf{0},\Sigma)$ distribution, where ij^{th} entry of Σ is $\rho^{|i-j|}$ and $\rho = 0.5$. The sample size is kept at 100 for all settings. In each setting, 10-fold cross-validation method was used to select the best λ for the lasso and the ridge regression. For the LINEX regression, we consider four values of α, $\alpha \in \{0.1, 0.5, 1, 5\}$. Then, for each fixed α, 10-fold cross-validation was used to select the best λ. Once we determined the best λ for each model, MSE was calculated with 5-fold cross validation keeping the folds fixed for each model in order to compare the performance of the LINEX models at each α, and also to compare the LINEX regression approach with other shrinkage approaches such as ridge regression and the lasso.

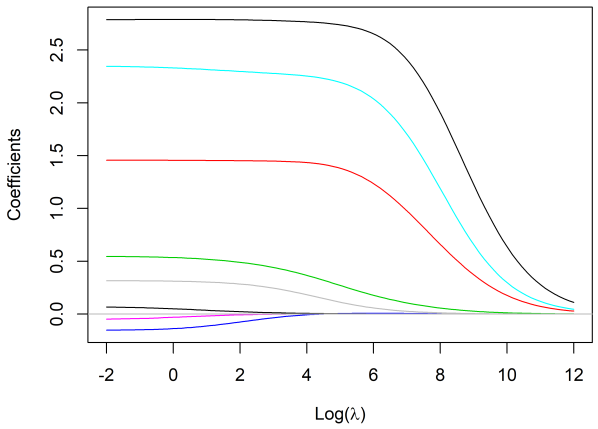
4.4.1 Sparse Setting

Similar to the sparse setting in Section 3.5, we set the true population coefficient vector to be $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^\top$. Figure 4.5 presents the trace plots for the LINEX regression models for different α. We observe that the behavior of LINEX regression is similar to the quadratic garrote behavior

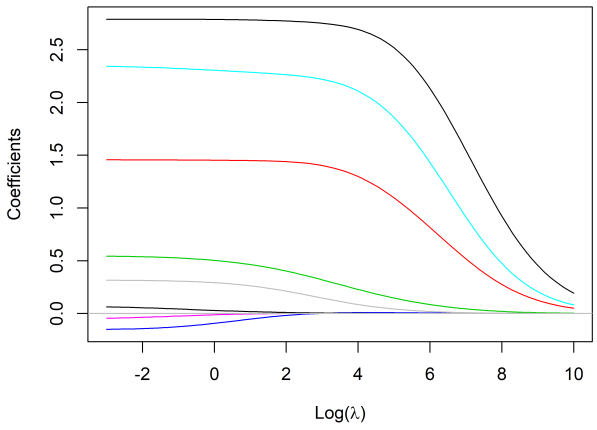
for smaller α values. Also, we see that solution becomes very sensitive to λ for large values of α. On the other hand, using a small α makes the LINEX coefficient estimates to be less sensitive to λ. Thus, a larger λ is required in order to shrink the coefficients when α is small. Figure 4.6 shows the cross-validation error plots for the LINEX regression models for different α. We see that in all the cases, there exists a λ (represented by the dotted line) which gives a lower prediction error than the OLS model. Furthermore, we notice that for a small α, the minimum prediction error is reached for a large value of λ, and vise versa.



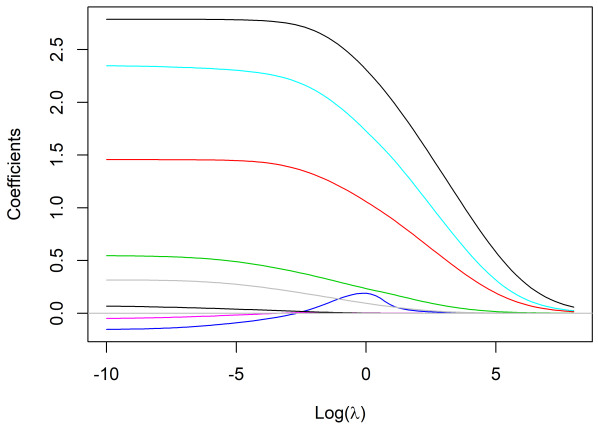
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

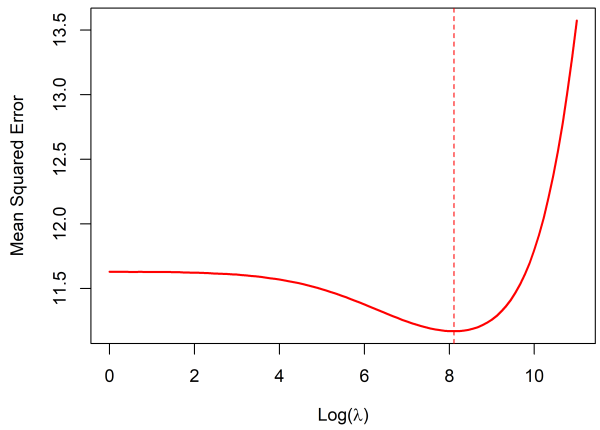


(c) $\alpha = 1$

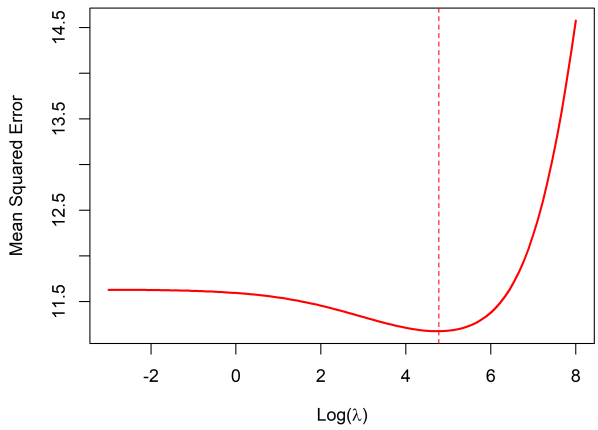


(d) $\alpha = 5$

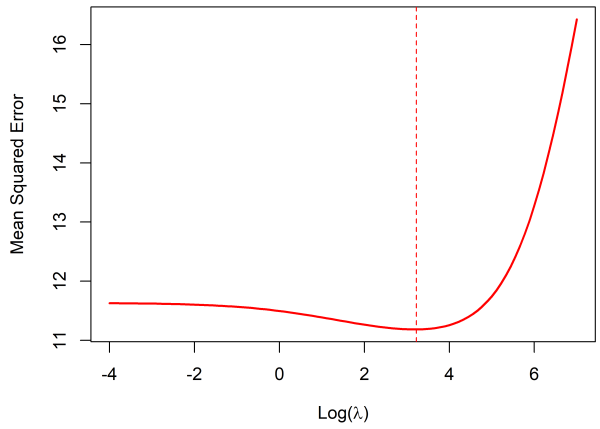
Figure 4.5: Trace plots of the LINEX regression models for different values of α under the sparse setting.



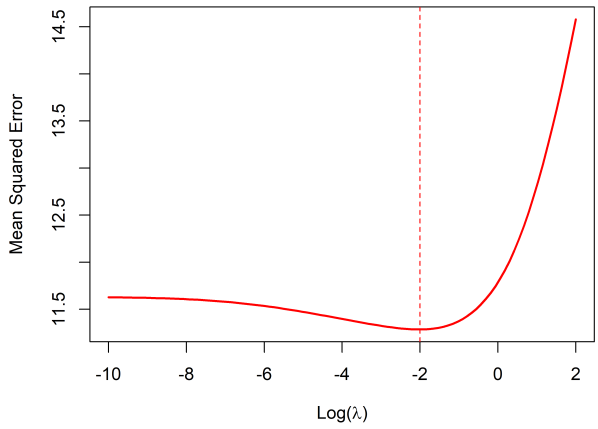
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

Figure 4.6: Cross-validation error plots of the LINEX regression models for different α under the sparse setting. Vertical dotted line in each plot represents the best λ.

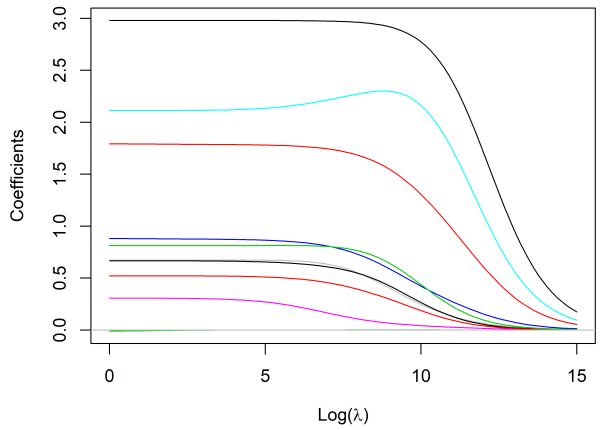
Table 4.1: Estimated coefficients and prediction errors under the sparse setting.

			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	3.0	2.786	0.861	2.572	2.743	2.788	2.587	2.667
X2	1.5	1.457	0.255	1.149	1.387	1.456	1.442	1.414
X3	0.0	0.547	0.019	0.164	0.300	0.538	0.504	0.381
X4	0.0	-0.155	0.002	0.009	0.005	-0.142	-0.023	0
X5	2.0	2.349	0.436	1.928	2.203	2.335	2.114	2.143
X6	0.0	-0.052	0.000	0.001	0.001	-0.036	-0.008	0
X7	0.0	0.316	0.003	0.051	0.133	0.313	0.300	0.213
X8	0.0	0.071	0.000	0.001	0.004	0.056	0.045	0
MSE		12.248	11.773	11.776	11.781	11.857	12.102	11.797

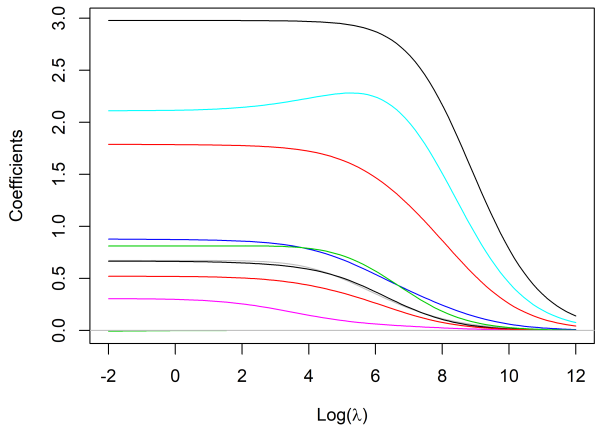
Table 4.1 summarizes the estimated coefficients and prediction errors for all the LINEX models along with the ridge and the lasso models. The prediction errors for all the LINEX models are very close to each other, and they are almost the same as the prediction error of the lasso. All LINEX regression models do better than the ridge regression in terms of the prediction accuracy. For $\alpha \in \{0.1, 0.5, 1\}$, all the LINEX regression coefficient estimates are positive, and all of them are smaller than the OLS estimates. When $\alpha = 0.1$, all the coefficient estimates, including the larger ones have been shrunk towards zero by a large amount. When $\alpha = 5$, the LINEX regression coefficients estimates are almost the same as the OLS estimates. When $\alpha = 0.5$ and $\alpha = 1$, LINEX regression approach has not shrunk the larger coefficients and had done more shrinkage on smaller coefficients.

4.4.2 Nearly-sparse Setting

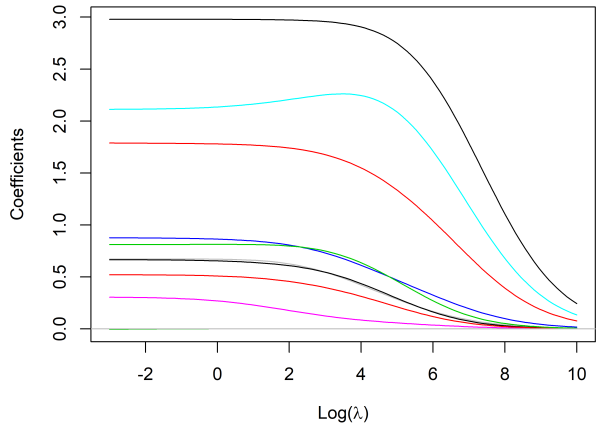
Similar to Section 3.5, in this setting we set the population coefficients vector to be $\boldsymbol{\beta} = (3, 1.5, z, z, 2, z, z, z, z, z)^\top$, where $z \sim Unif(0, b)$, $0 < b \leq 1$, and $b \in \{0.1, 0.5, 1\}$. The trace plots for LINEX regression under this setting for $b = 1$ are shown in Figure 4.7. The behavior of trace plots are very similar to those under the sparse setting. Cross-validation error plots of LINEX regression models for different α under the nearly-sparse setting for $b = 1$ are shown in Figure 4.7. By observing Figure 4.7, we can see that LINEX models do not show any significant drop in the prediction error curves, so that we can find a λ which minimizes the prediction error at each fixed α.



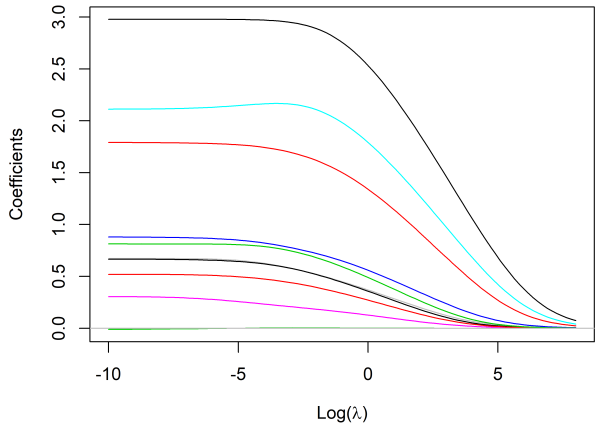
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

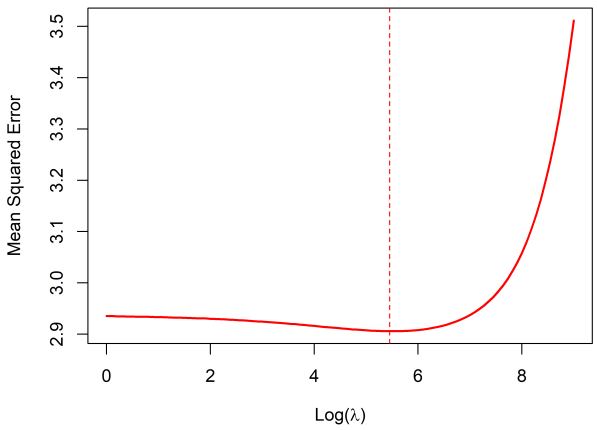


(c) $\alpha = 1$

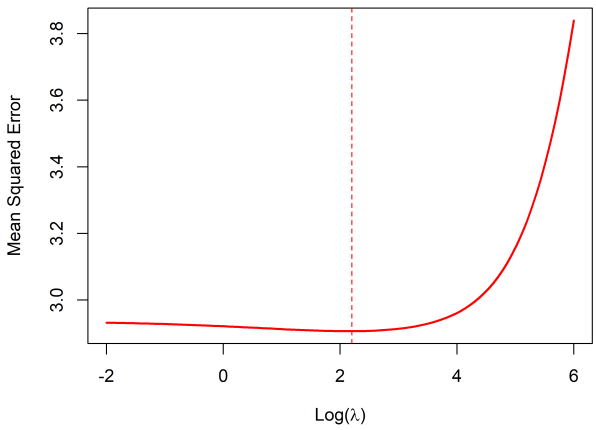


(d) $\alpha = 5$

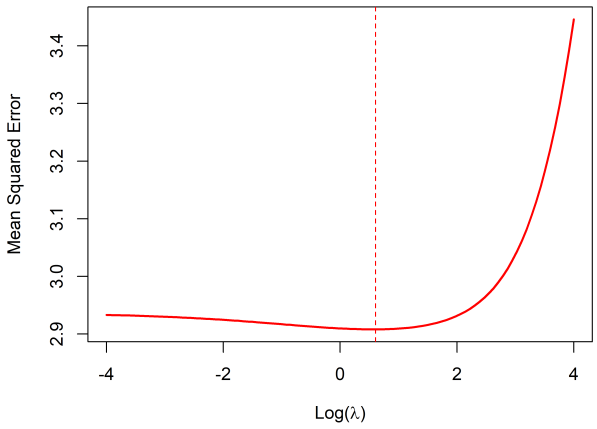
Figure 4.7: Trace plots of LINEX regression models for different α under the nearly-sparse setting for $b = 1$.



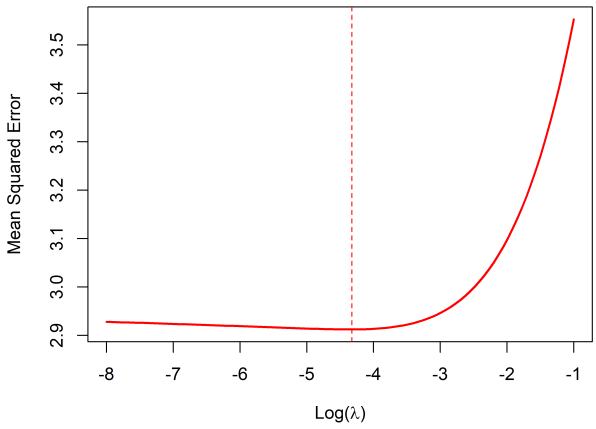
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

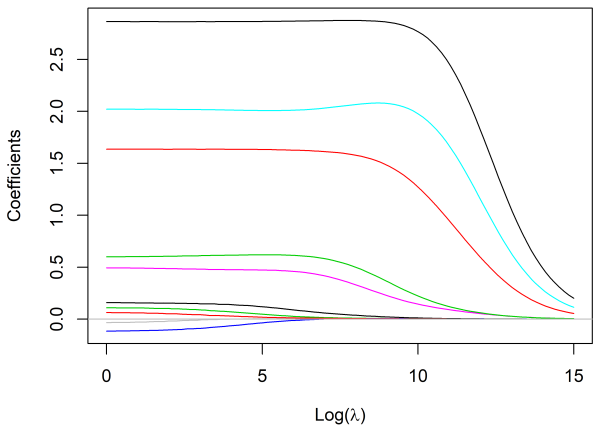
Figure 4.8: Cross-validation error plots of LINEX regression models for different α under the nearly-sparse setting for $b = 1$.

Table 4.2: Estimated coefficients and prediction errors under the nearly-sparse setting ($b = 1$).

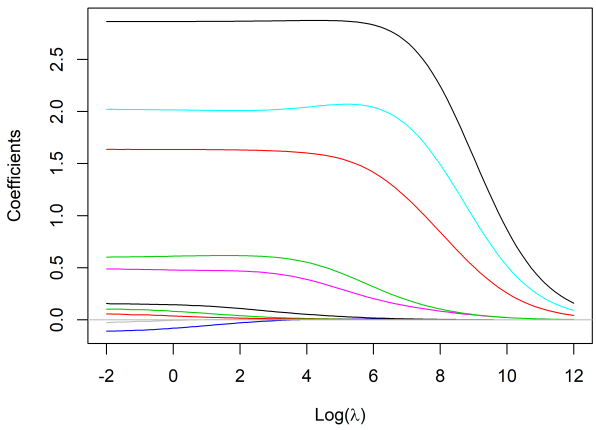
			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	3.000	2.978	2.613	2.971	2.978	2.978	2.809	2.942
X2	1.500	1.794	1.209	1.732	1.775	1.791	1.710	1.745
X3	0.197	-0.016	0.000	0.000	-0.000	-0.010	0.088	0
X4	0.716	0.882	0.403	0.796	0.856	0.879	0.847	0.838
X5	2.000	2.112	1.942	2.216	2.149	2.112	1.942	2.116
X6	0.362	0.305	0.048	0.167	0.250	0.305	0.420	0.276
X7	0.391	0.672	0.228	0.615	0.667	0.672	0.646	0.643
X8	0.813	0.666	0.231	0.600	0.647	0.666	0.648	0.637
X9	0.428	0.519	0.167	0.448	0.502	0.520	0.542	0.500
X10	0.959	0.813	0.372	0.793	0.812	0.813	0.735	0.778
MSE		3.083	3.066	3.066	3.067	3.078	3.075	3.033

Table 4.2 summarizes the estimated coefficients and prediction errors of each model when $b = 1$. All LINEX models has a better prediction accuracy than the ridge regression model for all the selected α. For $\alpha = 0.1$ and 0.5, we notice that the LINEX regression estimates of the smaller coefficients has been shrinked more than the larger coefficients an when For $\alpha = 1$ and 5, we do not notice a much shrinkage on any coefficient. Furthermore, all the LINEX regression estimates for the coefficient of X3 are very small. Note that those coefficients are not absolutely zero as in the lasso.

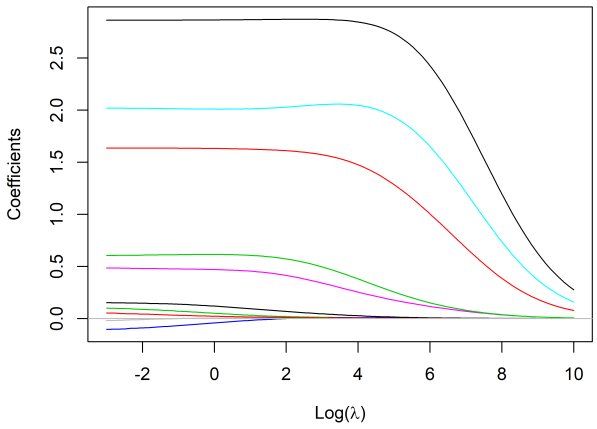
Consider the case when $b = 0.5$. The trace plots and the cross-validation error plots for LINEX regression under this setting are shown in Figure 4.9 and Figure 4.10 respectively. In this case as well, as we reduce α, the smaller coefficients shrink faster than the larger coefficients. In the trace plot of LINEX regression with $\alpha = 5$, we notice that one coefficient changes the sign from negative to positive and then shrinks to zero rapidly. Figure 4.10 clearly shows that, for all LINEX regression models, there exists a λ, which results in a lower prediction error than the OLS model.



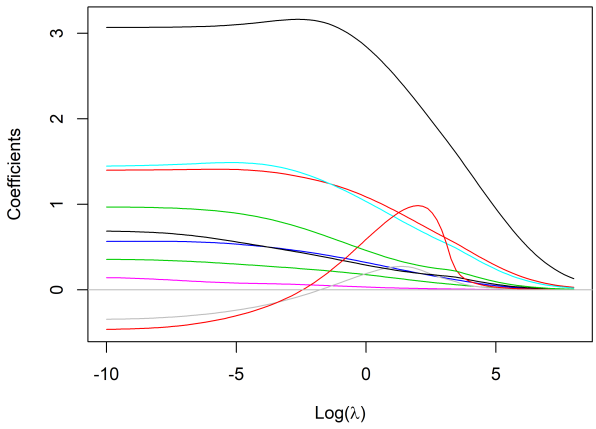
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

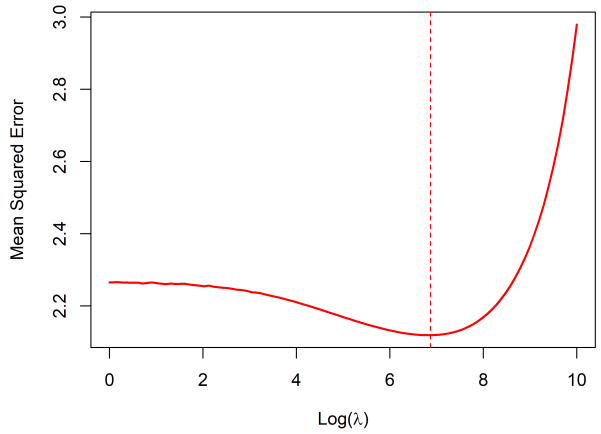


(c) $\alpha = 1$

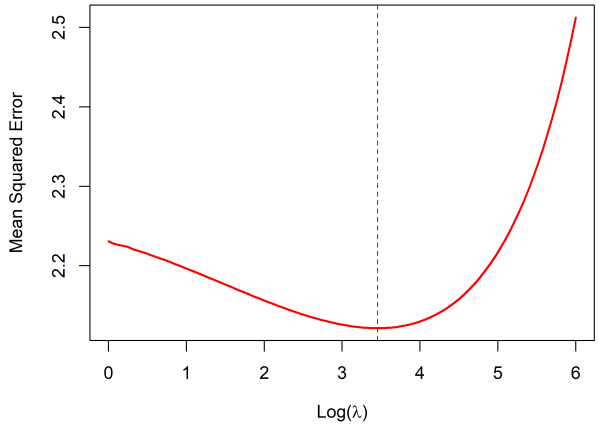


(d) $\alpha = 5$

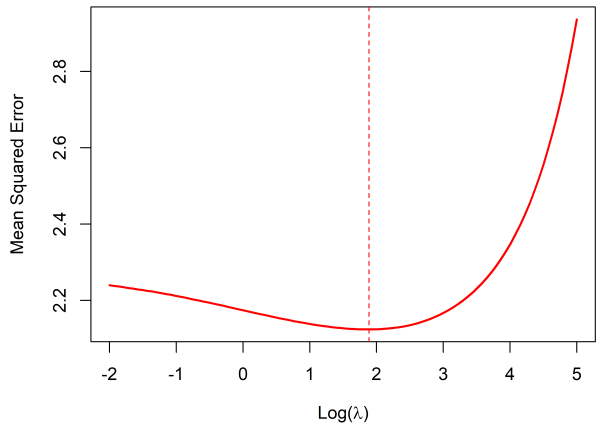
Figure 4.9: Trace plots (Left) and cross-validation error plots (Right) of LINEX regression models for different α under the nearly-sparse setting for $b = 0.5$.



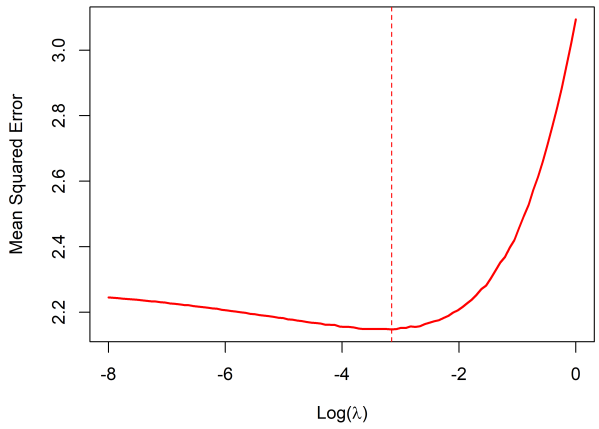
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

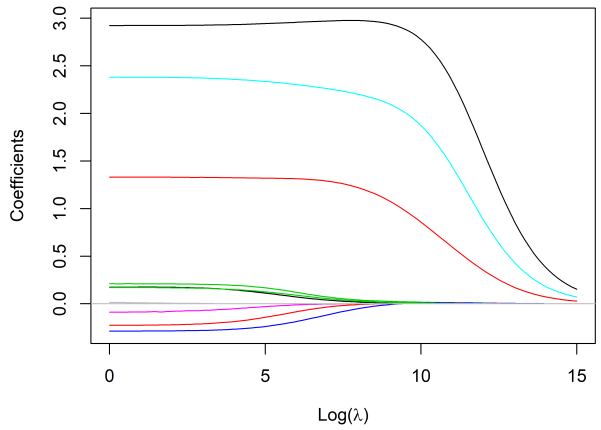
Figure 4.10: Cross-validation error plots of LINEX regression models for different α under the nearly-sparse setting for $b = 0.5$.

Table 4.3: Estimated coefficients and prediction errors under the nearly-sparse setting ($b = 0.5$).

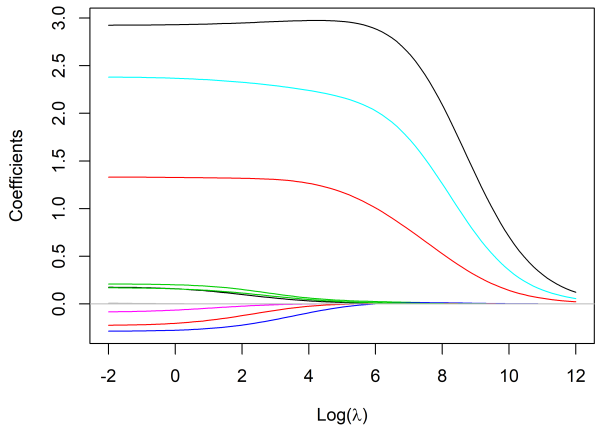
			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	3.000	2.864	1.955	2.865	2.871	2.864	2.658	2.833
X2	1.500	1.637	0.723	1.538	1.614	1.636	1.564	1.608
X3	0.154	0.111	0.003	0.011	0.021	0.101	0.190	0.054
X4	0.129	-0.117	0.004	0.009	-0.002	-0.105	0.079	0
X5	2.000	2.021	1.275	2.058	2.027	2.020	1.747	1.971
X6	0.276	0.495	0.076	0.302	0.421	0.487	0.576	0.457
X7	0.028	-0.037	0.000	0.002	0.004	-0.022	0.006	0
X8	0.234	0.160	0.004	0.037	0.072	0.155	0.134	0.147
X9	0.242	0.065	0.001	0.006	0.011	0.055	0.019	0.004
X10	0.406	0.599	0.088	0.447	0.579	0.603	0.599	0.594
MSE		2.337	2.216	2.215	2.215	2.215	2.377	2.220

Estimated coefficients and prediction errors for each model are presented in Table 4.3. In this case, all LINEX regression models show smaller prediction errors than the ridge regression model and the lasso model. In this case, ridge regression shows the highest prediction error. As we observe the estimated coefficients, at $\alpha = 0.1$, all estimated coefficients have been shrunk by a large proportion. The LINEX regression with $\alpha = 0.5$ or $\alpha = 1$ shrinks the smaller coefficients and does not shrink the larger ones. Using $\alpha = 5$ produce similar results to the OLS model.

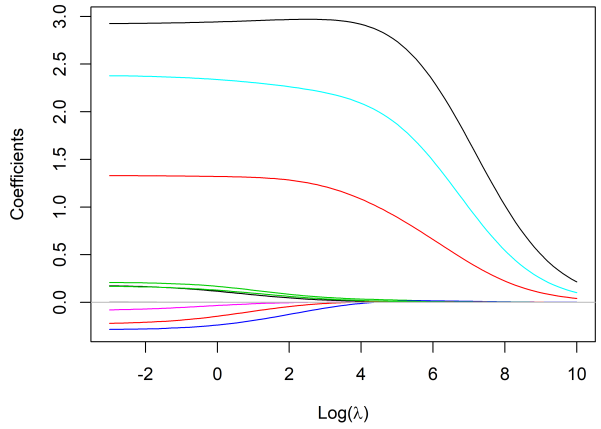
The third nearly sparse setting is for $b = 0.1$ which is expected to result in very small coefficients in the underlying true model. The trace plots and cross-validation error plots for LINEX regression models are shown in Figures 4.11 and 4.12, respectively. Except for few more smaller coefficients, we do not see any major difference in the behavior of these plots from the previous nearly-sparse case. Figure 4.12 shows the existence of a λ which minimizes the prediction error for all α values which we considered. Estimated coefficients along with the prediction error for each model are presented in Table 4.4. In this case, all LINEX regression models perform better than the ridge regression model and the lasso model with respect to the prediction error. The LINEX regression for $\alpha = 0.1$ gives all positive coefficients which is also the case for the true model. For $\alpha = 0.5$ and $\alpha = 1$, we see that the LINEX regression tends to shrink all small coefficients more than the ridge regression and large coefficient have not been shrunk much. In fact, when $\alpha = 1$, the largest coefficients have been grown in size than the OLS estimates.



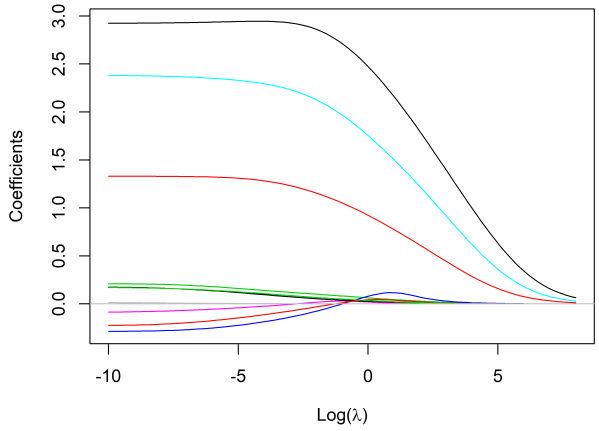
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

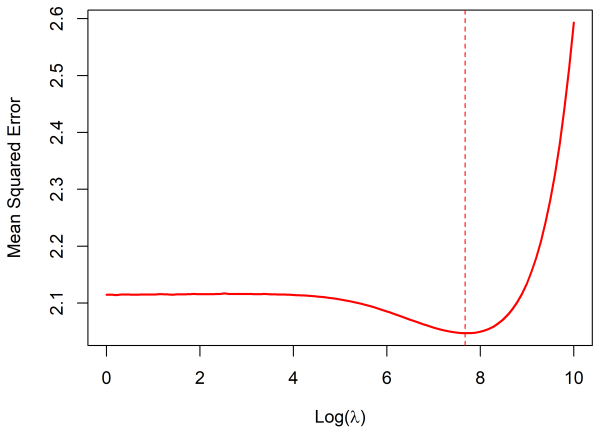


(c) $\alpha = 1$

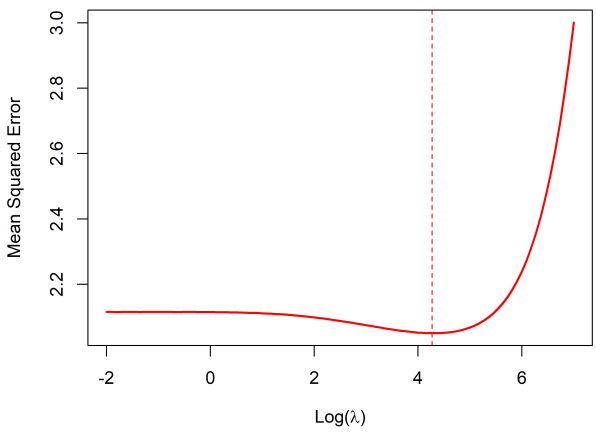


(d) $\alpha = 5$

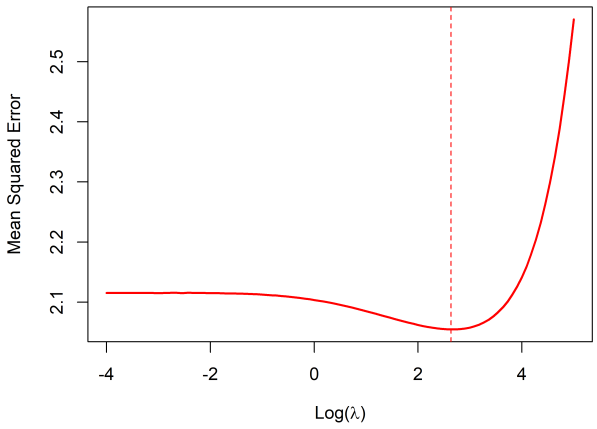
Figure 4.11: Trace plots of LINEX regression models for different α under the nearly-sparse setting for $b = 0.1$.



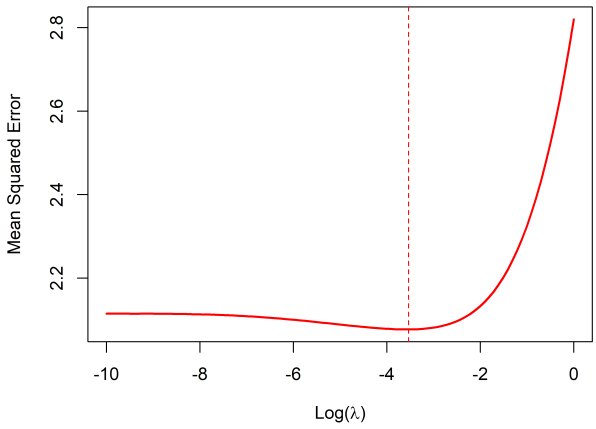
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

Figure 4.12: Trace plots (Left) and cross-validation error plots (Right) of LINEX regression models for different α under the nearly-sparse setting for $b = 0.1$

Table 4.4: Estimated coefficients and prediction errors under the nearly-sparse setting ($b = 0.1$).

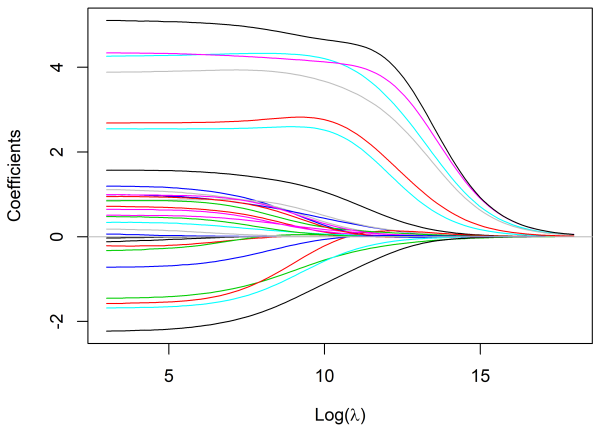
			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	3.000	2.924	1.226	2.886	2.970	2.925	2.693	2.882
X2	1.500	1.331	0.280	1.037	1.245	1.331	1.300	1.242
X3	0.034	0.211	0.006	0.031	0.063	0.209	0.271	0.063
X4	0.046	-0.288	0.007	-0.002	-0.083	-0.286	-0.173	0
X5	2.000	2.381	0.675	2.043	2.226	2.380	2.077	2.093
X6	0.006	-0.091	0.001	0.002	-0.000	-0.082	0.003	0
X7	0.020	0.018	0.000	0.000	0.000	0.008	0.044	0
X8	0.047	0.174	0.001	0.011	0.031	0.174	0.139	0
X9	0.030	-0.228	0.003	0.004	-0.021	-0.223	-0.147	0
X10	0.061	0.171	0.002	0.016	0.045	0.169	0.169	0.052
MSE		2.189	2.111	2.112	2.114	2.123	2.238	2.142

4.4.3 High Dimensional Setting

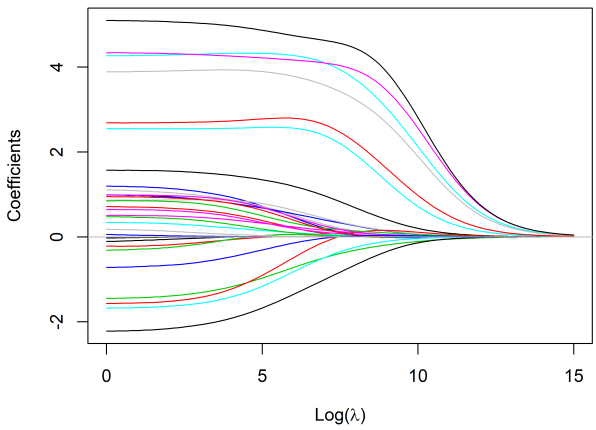
Now, we evaluate the performance our model in a high dimensional setting. To this end, we use the high dimensional setting as in Section 3.5. Recall the summary of the three scenarios

- Scenario 1: Population coefficient vector consists of the majority of very small coefficients and some relatively large coefficients,
- Scenario 2: Population coefficient vector consists of equal numbers of moderately large coefficients and small coefficients,
- Scenario 3: Population coefficient vector contains many small coefficients and some moderately large coefficients with few very large coefficients.

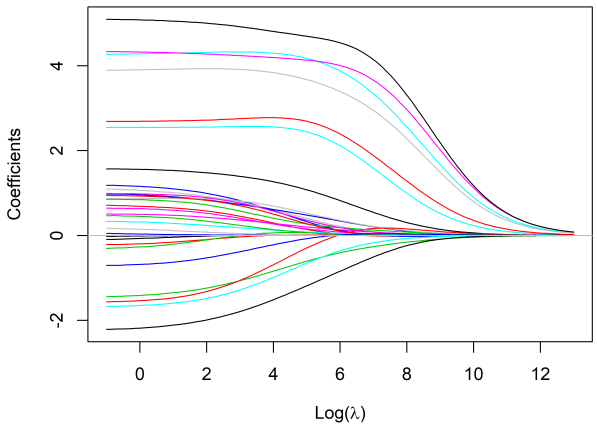
Trace plots and cross-validation error plots of LINEX regression models for different α under the high-dimensional setting for Scenario 1 are given in Figures 4.13 and 4.14, respectively. Here as well we observe the same behavior in the trace plots and cross-validation error plots we saw in the previous cases. The estimated coefficients and prediction errors under this scenario are given in Table 4.5. We see that the prediction errors of all LINEX regression models are higher than the ridge regression model. For this scenario, ridge regression gives even better prediction accuracy than the lasso. LINEX regression model with $\alpha = 0.1$ results in coefficients that are very smaller in size than the OLS estimates. On the other hand, using $\alpha = 5$ does not do much shrinkage. Considering the amount of shrinkage on the coefficients and the prediction accuracy, one can suggest $\alpha = 0.5$ or $\alpha = 1$ for this setting.



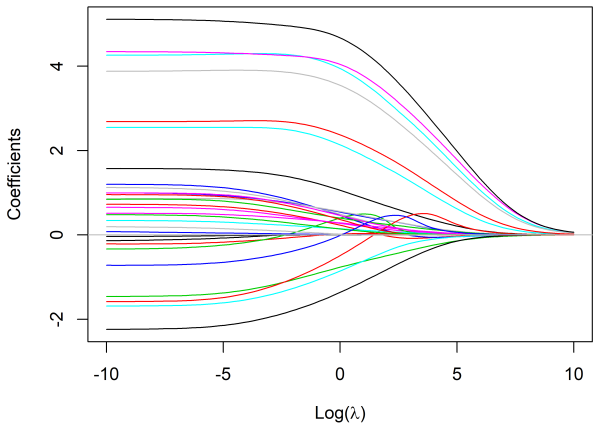
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

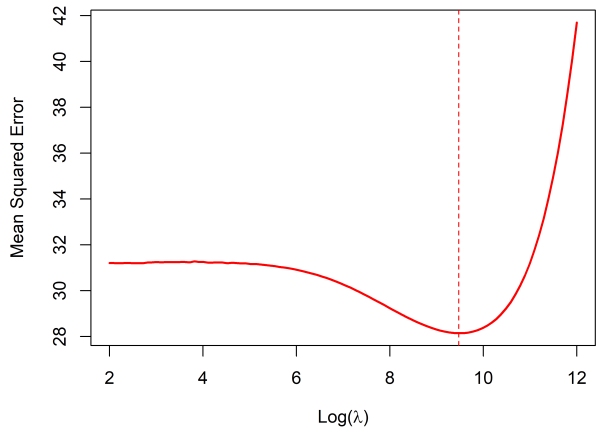


(c) $\alpha = 1$

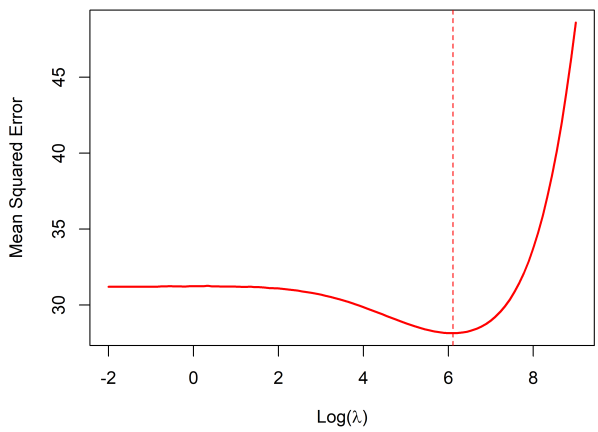


(d) $\alpha = 5$

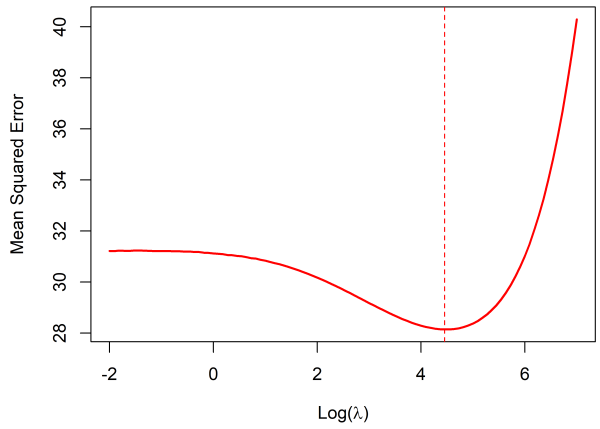
Figure 4.13: Trace plots of LINEX regression models for different α under the high-dimensional setting (Scenario 1).



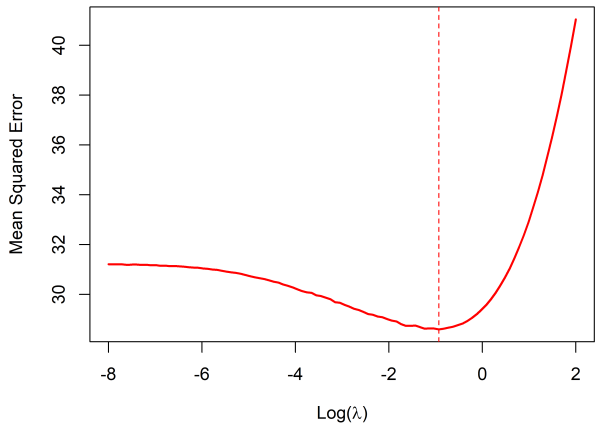
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

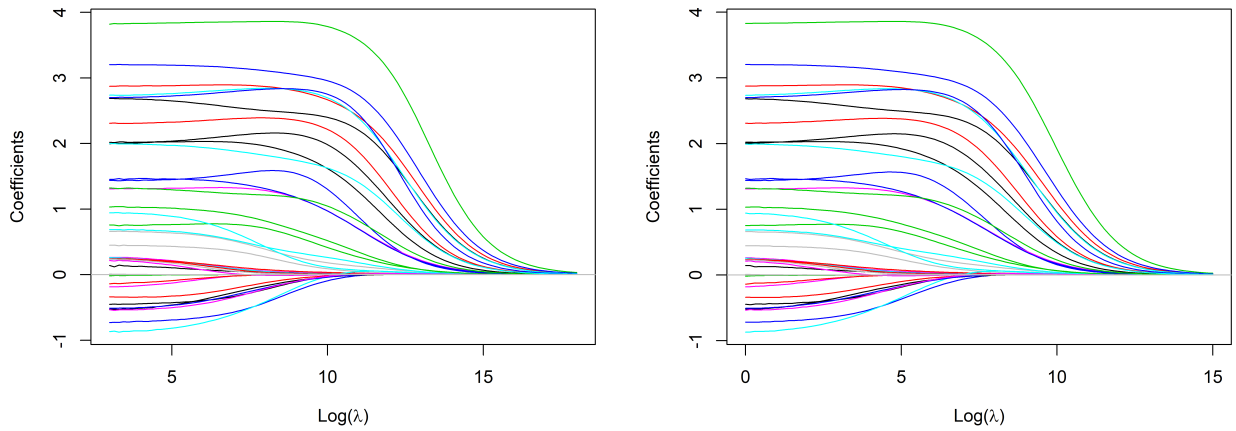
Figure 4.14: Cross-validation error plots of LINEX regression models for different α under the high-dimensional setting (Scenario 1).

Table 4.5: Estimated coefficients and prediction errors under the high-dimensional setting (Scenario 1).

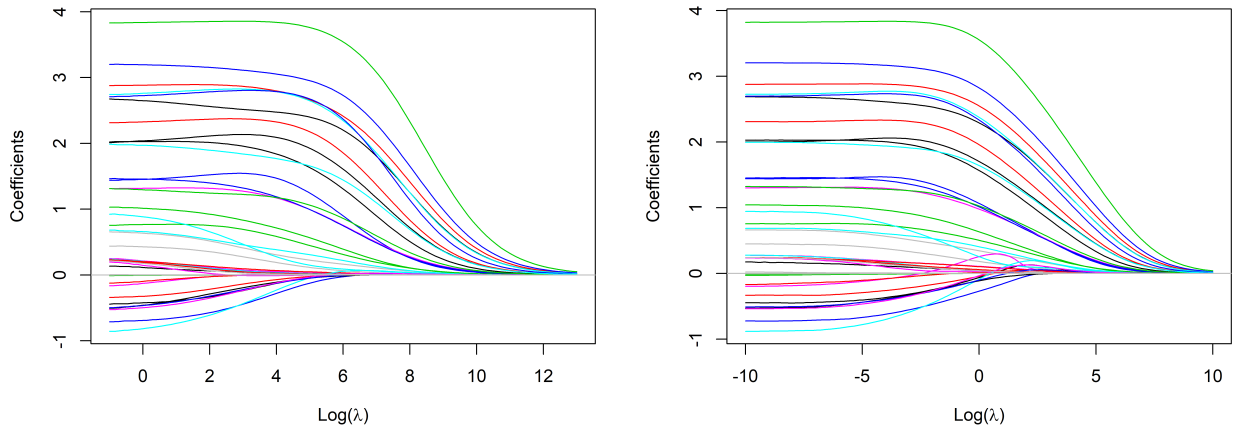
			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	0.025	-0.041	-0.000	-0.000	-0.000	-0.020	-0.226	0
X2	0.008	0.729	-0.010	0.031	0.223	0.707	0.330	0
X3	0.071	-1.459	-0.058	-0.387	-0.731	-1.438	-0.969	-0.267
X4	0.248	1.201	-0.015	0.105	0.467	1.184	0.591	0
X5	0.033	-1.688	-0.015	-0.332	-0.835	-1.669	-0.964	0
X6	0.088	0.656	0.007	0.078	0.215	0.643	0.638	0
X7	0.077	1.124	0.005	0.197	0.517	1.092	0.721	0.217
X8	0.060	-0.137	-0.000	0.002	0.007	-0.094	-0.147	0
X9	0.139	-0.215	0.001	0.011	0.013	-0.215	0.029	0
X10	0.063	-0.333	0.003	0.040	0.070	-0.298	-0.118	0.060
X11	1.000	0.997	0.030	0.326	0.556	0.978	0.725	0.357
X12	1.000	2.547	0.362	2.067	2.528	2.549	1.987	2.106
X13	1.000	0.516	0.014	0.128	0.247	0.504	0.774	0.438
X14	1.000	0.857	0.037	0.339	0.621	0.866	1.043	1.081
X15	1.000	1.575	0.097	0.792	1.190	1.568	1.034	0.785
X16	0.254	-1.582	0.068	0.044	-0.506	-1.560	-0.746	0
X17	0.070	0.846	0.031	0.178	0.340	0.851	0.673	0.074
X18	0.256	0.076	0.000	0.001	0.004	0.047	-0.004	0
X19	0.276	0.341	0.005	0.048	0.108	0.335	0.649	0.467
X20	0.263	0.996	0.007	0.170	0.477	0.988	0.623	0
X21	0.070	0.196	-0.000	0.001	0.016	0.169	-0.142	0
X22	0.048	-2.241	-0.051	-0.801	-1.367	-2.210	-1.367	-0.518
X23	0.168	0.957	-0.001	0.091	0.402	0.942	0.578	0
X24	0.033	0.478	0.003	0.025	0.104	0.471	0.324	0
X25	0.080	-0.720	0.017	0.019	-0.146	-0.708	-0.116	0
X26	4.000	4.260	1.274	3.830	4.258	4.272	3.788	4.036
X27	4.000	4.344	1.563	3.982	4.169	4.333	3.870	3.884
X28	4.000	3.879	1.134	3.353	3.776	3.893	3.648	3.611
X29	4.000	5.112	1.666	4.515	4.758	5.093	4.225	4.137
X30	4.000	2.684	0.532	2.345	2.759	2.689	2.776	3.088
MSE		33.863	32.614	32.507	32.384	31.504	29.639	30.615

Under Scenario 2, trace plots and cross-validation error plots of LINEX regression models are given in Figures 4.15 and 4.16, respectively. The estimated coefficients and prediction errors are given in Table 4.6. In this case, the lasso performs the best in terms of the prediction error. The ridge regression performs slightly better than LINEX regression models. Here as well, LINEX regression model with $\alpha = 0.1$ gives a large shrinkage on all coefficient estimates even for the larger coefficients, and on the other hand working with LINEX regression model with $\alpha = 5$ results in similar coefficient estimates to

OLS estimates. Hence, for this case as well, $\alpha = 0.5$ or $\alpha = 1$ can be recommended to perform LINEX regression.

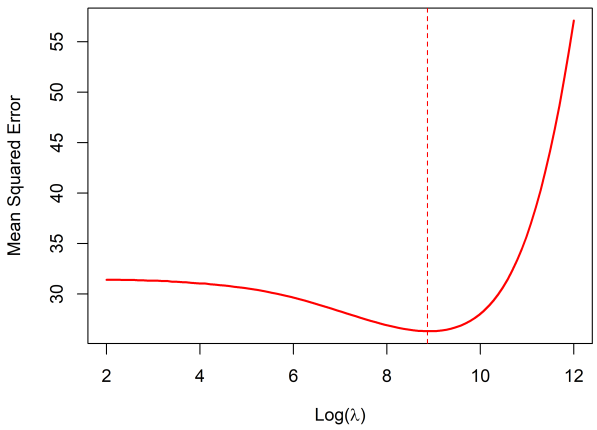


(a) $\alpha = 0.1$ (b) $\alpha = 0.5$

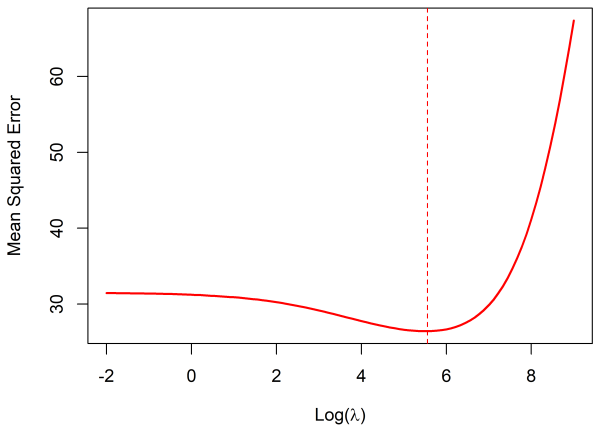


(c) $\alpha = 1$ (d) $\alpha = 5$

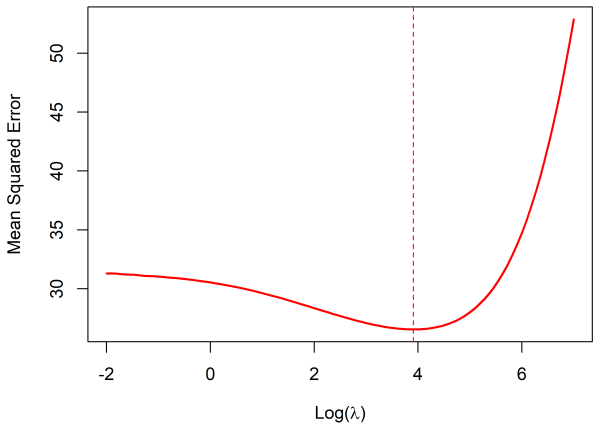
Figure 4.15: Trace plots of LINEX regression models for different α under the high-dimensional setting (Scenario 2).



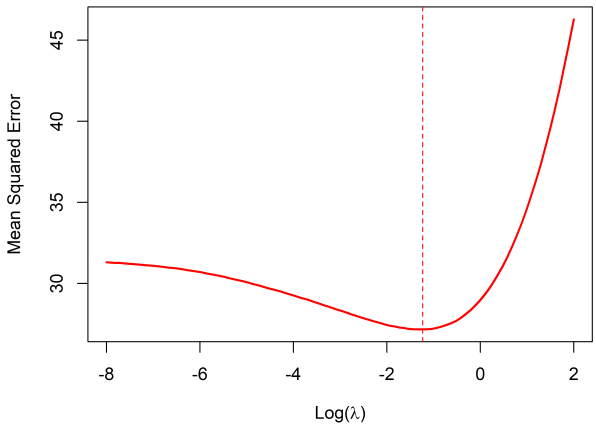
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



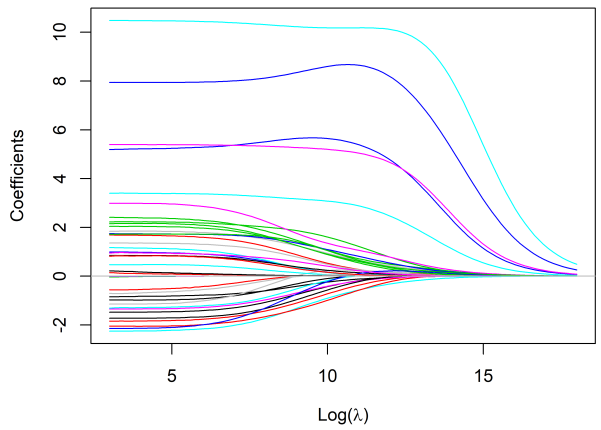
(d) $\alpha = 5$

Figure 4.16: Cross-validation error plots (Right) of LINEX regression models for different α under the high-dimensional setting (Scenario 2).

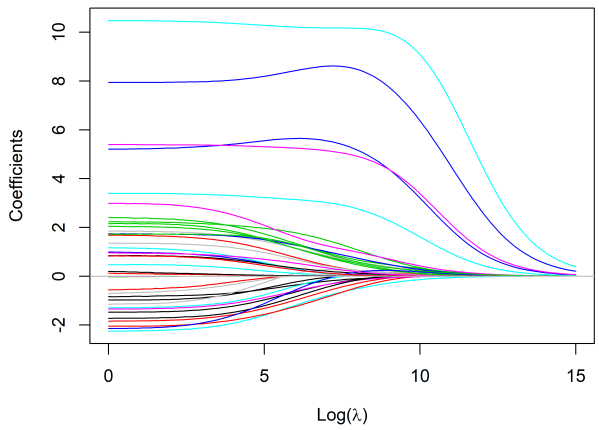
Table 4.6: Estimated coefficients and prediction errors under the high-dimensional setting (Scenario 2).

			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	3.000	2.688	0.778	2.311	2.479	2.679	2.297	2.228
X2	3.000	2.877	0.900	2.542	2.807	2.877	2.770	2.806
X3	3.000	3.820	1.570	3.664	3.843	3.831	3.307	3.740
X4	3.000	3.204	1.070	2.850	3.060	3.200	2.761	2.859
X5	3.000	2.723	0.767	2.503	2.793	2.738	2.186	2.669
X6	0.021	0.278	0.007	0.034	0.053	0.258	0.687	0
X7	0.090	0.662	0.032	0.190	0.316	0.657	0.647	0.362
X8	0.059	-0.446	0.009	-0.007	-0.138	-0.447	-0.293	0
X9	0.075	-0.333	0.004	-0.008	-0.075	-0.337	0.043	0
X10	0.017	-0.033	0.000	-0.000	-0.001	-0.012	0.012	0
X11	0.086	-0.725	0.006	-0.061	-0.295	-0.714	-0.623	-0.073
X12	0.028	0.280	0.002	0.017	0.036	0.245	0.159	0
X13	0.057	-0.199	0.002	0.015	0.016	-0.171	0.039	0
X14	0.057	0.447	0.008	0.085	0.197	0.443	0.416	0.406
X15	0.037	0.179	0.001	0.012	0.028	0.133	0.171	0
X16	1.000	-0.173	0.000	0.004	0.002	-0.126	-0.157	0
X17	1.000	1.044	0.050	0.472	0.726	1.025	0.832	0.497
X18	1.000	1.439	0.122	1.061	1.483	1.446	1.193	1.619
X19	1.000	0.688	0.032	0.260	0.390	0.681	0.741	0.307
X20	1.000	1.301	0.148	0.867	1.183	1.308	1.183	1.054
X21	1.000	0.022	0.000	0.000	0.001	0.006	0.766	0.537
X22	1.000	2.000	0.404	1.766	2.100	2.008	1.410	1.841
X23	1.000	0.237	0.005	0.038	0.070	0.236	0.607	0.292
X24	1.000	0.755	0.055	0.371	0.596	0.757	0.928	0.906
X25	1.000	1.453	0.137	0.862	1.204	1.460	1.062	0.896
X26	0.169	-0.887	0.020	0.024	-0.248	-0.859	-0.306	0
X27	0.017	0.242	0.002	0.002	-0.011	0.205	0.388	0
X28	0.026	-0.016	0.000	-0.000	-0.000	-0.004	-0.093	0
X29	0.174	-0.532	0.001	-0.024	-0.116	-0.514	-0.329	0
X30	0.076	0.244	0.002	0.012	0.026	0.233	0.186	0
X31	0.002	-0.007	0.000	0.000	0.000	-0.002	0.306	0
X32	0.119	-0.514	0.008	-0.019	-0.133	-0.504	-0.288	0
X33	0.070	0.945	0.017	0.110	0.293	0.933	0.280	0
X34	0.120	-0.538	0.011	-0.001	-0.129	-0.532	0.048	0
X35	0.009	0.242	0.001	0.010	0.017	0.223	0.144	0
X36	2.000	2.029	0.265	1.466	1.853	2.018	1.788	1.765
X37	2.000	2.307	0.460	2.022	2.333	2.310	1.998	2.258
X38	2.000	1.322	0.181	0.959	1.184	1.321	1.475	1.288
X39	2.000	2.695	0.640	2.536	2.792	2.705	1.947	2.518
X40	2.000	1.994	0.394	1.555	1.776	1.991	2.069	1.752
MSE		35.550	25.984	26.082	26.412	28.283	25.866	21.523

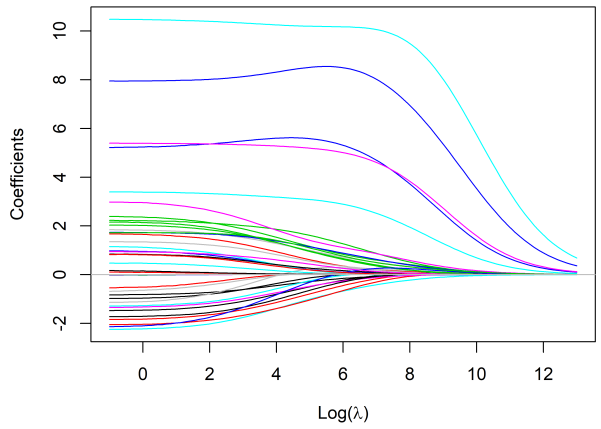
Now, consider the Scenario 3, where we have many small coefficients with some moderately large and few large coefficients in the true regression model. The trace plots for LINEX regression models are presented in Figure 4.17. We see that for all values of α which we considered, the small coefficients shrink faster than the large coefficients. For $\alpha = 5$, some small coefficients change the sign as λ increases. Figure 4.18 presents the cross-validation error plots under Scenario 3. All LINEX regression models show a considerable reduction in the prediction error for some λ. When we examine the prediction errors in Table 4.7, we see that all LINEX regression models perform better than the ridge regression and also, except for $\alpha = 5$, LINEX regression approach provides competitive prediction errors as the lasso. In this case, LINEX regression model, with the lowest α gives a slightly better prediction accuracy than LINEX regression model using a large α. However, a small α can yield unnecessarily small coefficients as we see with LINEX model with $\alpha = 0.1$. On the other hand, LINEX regression model with $\alpha = 5$ results in a high prediction error. Hence, in this case as well, we recommend using $\alpha = 0.5$ or $\alpha = 1$ to maintain some trade-off between the prediction accuracy and the shrinkage under this setting. Furthermore, we notice that the OLS model has a considerably higher prediction error than all the considered shrinkage methods. Hence, we can safely conclude that the use of a shrinkage method rather than OLS model in the high dimensional setting is essential to have a better prediction accuracy.



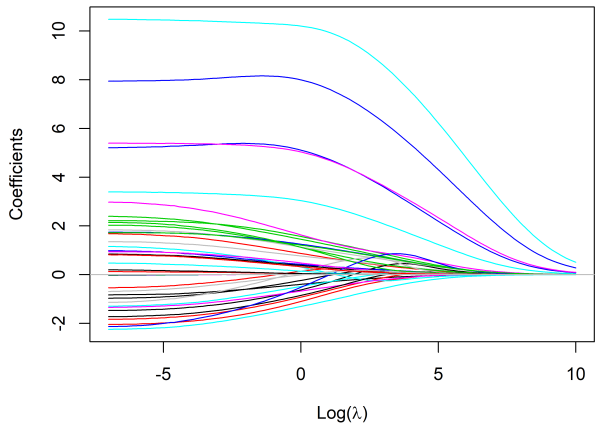
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

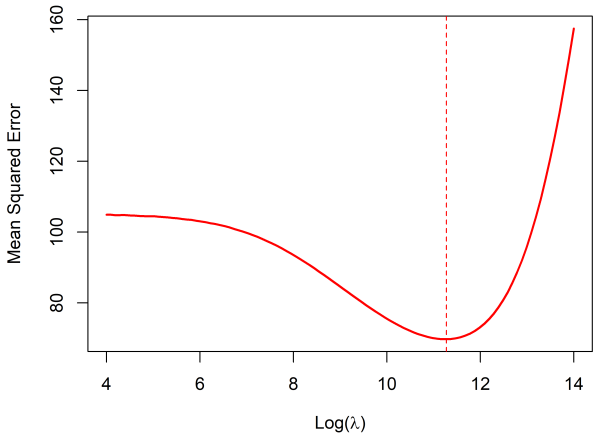


(c) $\alpha = 1$

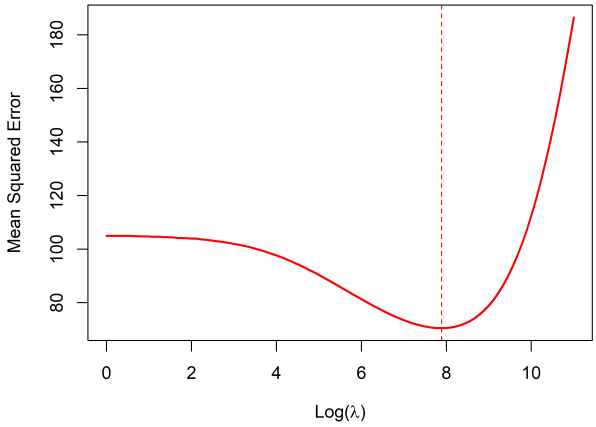


(d) $\alpha = 5$

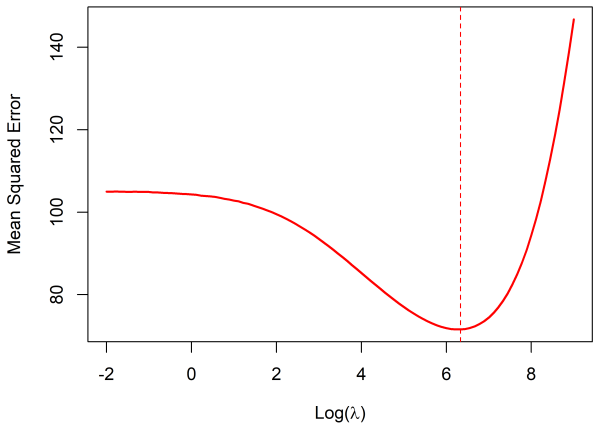
Figure 4.17: Trace plots of LINEX regression models for different α under the high-dimensional setting (Scenario 3).



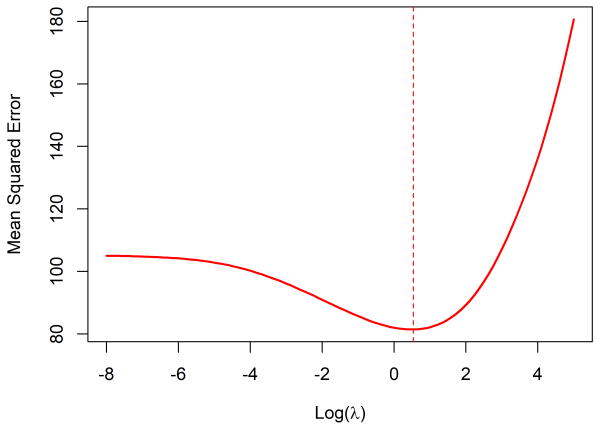
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

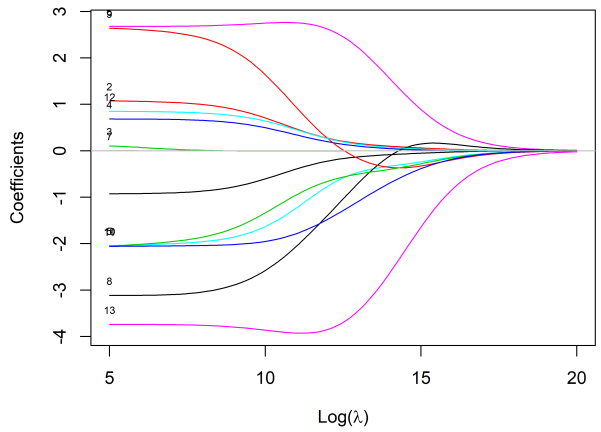
Figure 4.18: Cross-validation error plots of LINEX regression models for different α under the high-dimensional setting (Scenario 3).

Table 4.7: Estimated coefficients and prediction errors under the high-dimensional setting (Scenario 3).

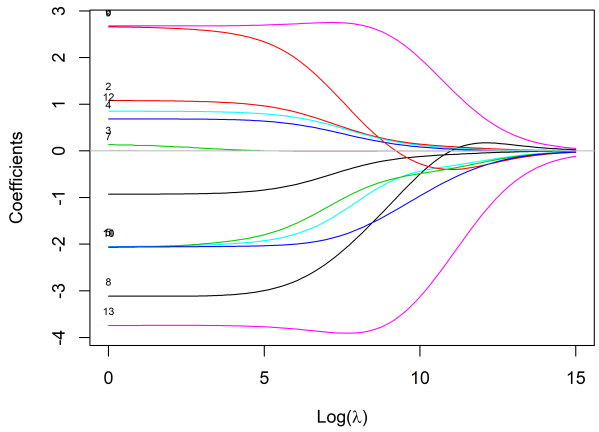
			LINEX					
	β_j	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
X1	0.008	-0.973	0.004	0.035	0.024	-0.943	-0.235	0
X2	0.032	1.705	0.016	0.134	0.295	1.623	1.032	0
X3	0.249	-0.042	0.000	0.000	0.000	-0.005	0.020	0
X4	0.136	1.004	0.008	0.051	0.088	0.934	0.809	0
X5	0.154	-2.261	-0.011	-0.236	-0.577	-2.192	-1.536	0
X6	0.124	-0.018	-0.000	-0.000	-0.000	-0.003	-0.252	0
X7	0.126	0.860	0.001	0.001	0.040	0.816	0.872	0
X8	0.115	-0.838	-0.002	-0.035	-0.120	-0.798	-0.863	0
X9	0.080	-2.050	0.004	-0.083	-0.522	-2.014	-0.967	0
X10	0.046	2.419	0.038	0.437	0.747	2.328	1.330	0
X11	5.000	5.185	0.515	3.841	5.146	5.263	4.913	5.148
X12	5.000	3.401	0.249	2.019	2.792	3.379	3.411	3.348
X13	5.000	5.400	0.612	3.937	4.907	5.385	4.526	4.763
X14	0.064	1.364	0.022	0.174	0.300	1.331	1.241	0
X15	0.041	-1.483	0.019	0.049	-0.154	-1.435	-0.489	0
X16	0.166	-1.846	0.016	-0.019	-0.340	-1.797	-1.137	0
X17	0.003	2.041	0.017	0.203	0.527	1.998	1.186	0
X18	0.003	-0.004	-0.000	0.000	0.000	-0.001	0.357	0
X19	0.366	1.182	-0.005	-0.020	0.027	1.090	-0.004	0
X20	0.101	-1.352	0.001	-0.043	-0.201	-1.306	-0.769	0
X21	0.011	1.846	0.041	0.354	0.644	1.816	1.396	0.415
X22	0.026	-1.737	0.010	0.016	-0.205	-1.683	-0.856	0
X23	0.118	-0.557	0.001	0.019	0.044	-0.484	-0.029	0
X24	1.000	2.221	0.039	0.552	1.137	2.202	1.940	1.377
X25	1.000	1.757	0.041	0.420	0.799	1.719	2.054	1.176
X26	1.000	-1.308	0.006	0.051	0.016	-1.241	-0.826	0
X27	1.000	2.997	0.059	0.615	1.033	2.908	1.666	0.299
X28	1.000	-1.160	0.006	0.102	0.260	-1.058	-0.076	0.359
X29	0.428	0.846	0.004	0.057	0.142	0.814	0.767	0.172
X30	0.031	0.158	0.000	0.002	0.008	0.083	0.332	0.372
X31	0.219	2.149	0.003	0.214	0.570	2.115	0.732	0
X32	0.130	-2.158	0.027	0.257	0.160	-2.052	-0.450	0
X33	0.140	0.479	0.000	0.008	0.029	0.457	0.229	0
X34	0.020	0.953	0.004	0.066	0.211	0.925	0.510	0.305
X35	0.023	-0.723	0.003	0.017	0.043	-0.601	0.113	0
X36	0.121	0.232	0.000	0.001	0.006	0.129	0.341	0
X37	0.020	0.820	0.002	0.015	0.068	0.799	0.123	0
X38	0.084	1.719	0.044	0.337	0.665	1.725	2.269	0.850
X39	10.000	7.940	1.595	7.089	8.396	7.949	7.332	8.485
X40	10.000	10.489	2.811	9.586	10.154	10.453	8.991	9.751
MSE		101.818	71.246	71.814	72.720	80.175	84.430	68.933

4.5 Example 1: The Boston Housing Dataset (Continued)

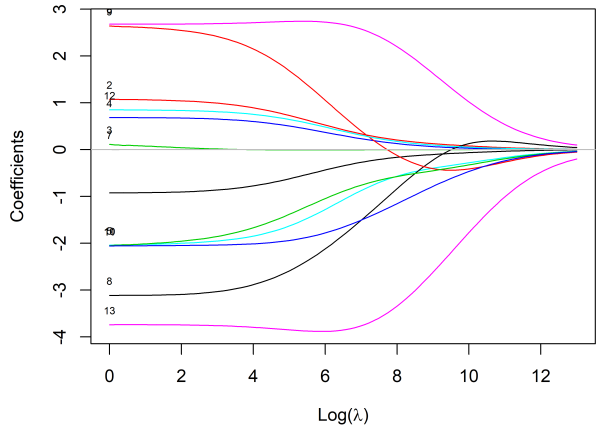
After the simulation study, once again consider the Boston housing dataset. Similar to the previous examples, suppose that we want to predict medv with all the other variables as predictors. We apply LINEX regression with the same set of α values which we used earlier in the simulation study section. The trace plots for each α are presented in Figure 4.19. We see that the largest positive coefficient (rm) and the largest (in size) negative coefficient (lstat) shrink slower than the all other coefficients. We also notice that the two coefficient estimates of the variables,rad and dis shrinks faster than the other coefficients as we increase λ. As we increase λ further, those coefficients even change their signs. Figure 4.20 presents the 10-fold cross-validation error plots for different α values. Table 4.8 summarizes the results of the considered shrinkage approaches. When we compare the prediction accuracies of the models, all LINEX regression models perform better than the ridge regression. Except for $\alpha = 5$, all the other LINEX regression models give better prediction errors than the lasso as well. The LINEX model with $\alpha = 5$ does not do any shrinkage in any coefficient in this case (not visible at two decimal places). For $\alpha = 0.1$, all the LINEX regression coefficients are smaller than the OLS estimates. For $\alpha = 0.5$ and $\alpha = 1$, LINEX regression coefficients has not shrunken the larger coefficients. In fact, the large coefficients are larger than the corresponding OLS estimates in size.



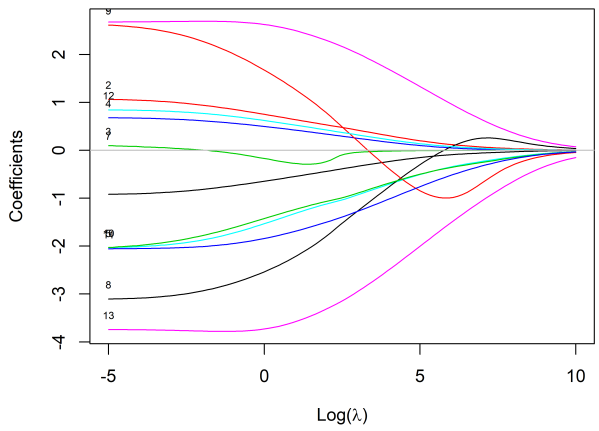
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

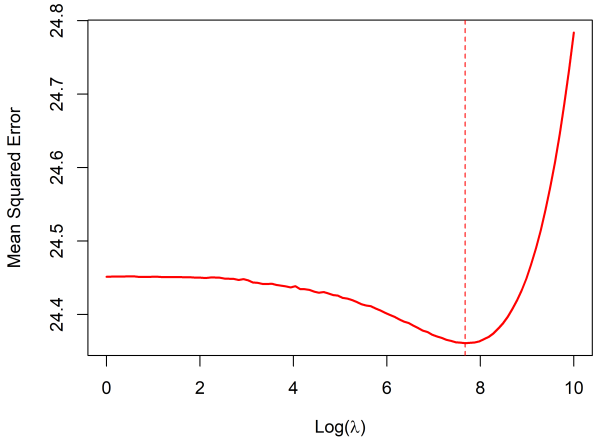


(c) $\alpha = 1$

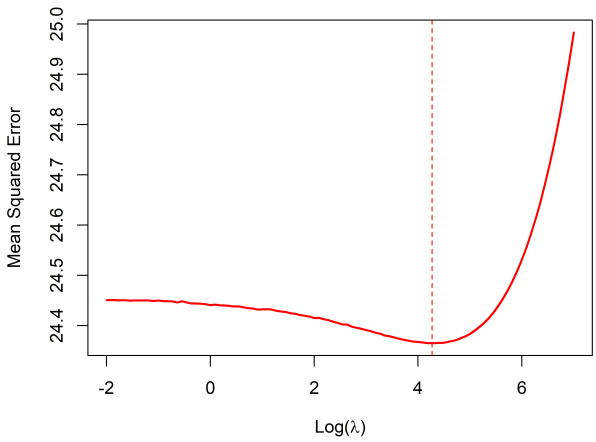


(d) $\alpha = 5$

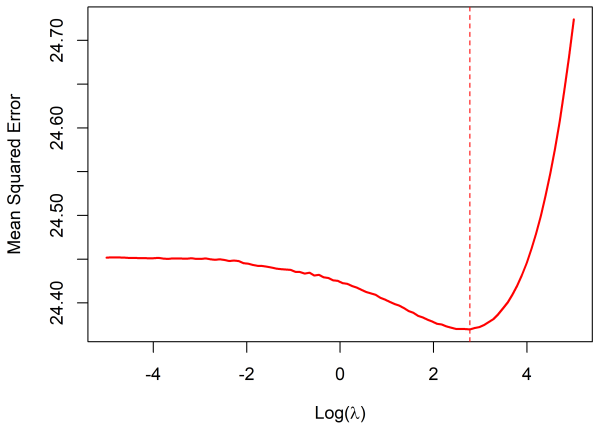
Figure 4.19: Trace plots of LINEX regression models for different α for the Boston housing dataset.



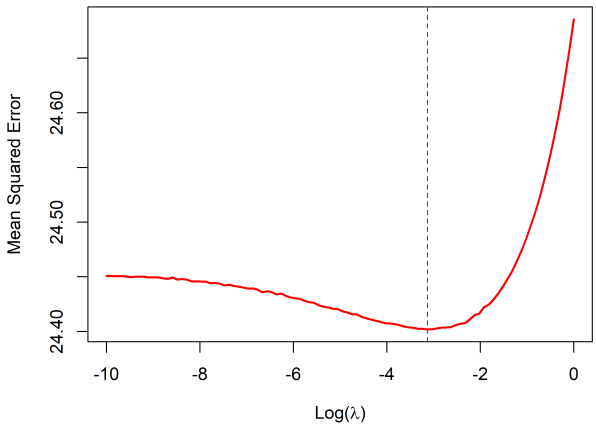
(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



(c) $\alpha = 1$



(d) $\alpha = 5$

Figure 4.20: Cross-validation error plots of LINEX regression models for different α for the Boston housing dataset.

Table 4.8: Estimated coefficients and prediction errors for the Boston housing dataset.

		LINEX					
	OLS	($\alpha = 0.1$)	($\alpha = 0.5$)	($\alpha = 1$)	($\alpha = 5$)	Ridge	Lasso
crim	-0.9291	-0.1926	-0.7390	-0.8755	-0.9289	-0.7441	-0.8818
zn	1.0826	0.2420	0.8559	1.0080	1.0824	0.7450	1.0093
indus	0.1410	-0.0043	-0.0046	0.0184	0.1395	-0.2765	0
chas	0.6824	0.1583	0.5790	0.6588	0.6825	0.7381	0.6859
nox	-2.0588	-0.6502	-1.8015	-1.9657	-2.0586	-1.3396	-1.9416
rm	2.6769	2.3466	2.7189	2.6879	2.6767	2.8215	2.6981
age	0.0195	-0.0000	-0.0001	0.0002	0.0200	-0.1113	0
dis	-3.1071	-1.0539	-2.8216	-3.0517	-3.1072	-2.3029	-3.0252
rad	2.6649	0.0236	2.0453	2.4547	2.6640	1.2771	2.3694
tax	-2.0788	-0.6243	-1.6018	-1.8829	-2.0776	-0.9274	-1.7925
ptratio	-2.0626	-1.2730	-1.9994	-2.0414	-2.0626	-1.8367	-2.0284
black	0.8501	0.2117	0.7305	0.8176	0.8500	0.8258	0.8343
lstat	-3.7473	-3.5162	-3.8085	-3.7554	-3.7474	-3.3445	-3.7315
MSE	23.292	23.223	23.227	23.232	23.257	23.690	23.252

When we compare the LINEX regression approach with other approaches, we can conclude that the LINEX regression approach is a worthy rival for ridge regression and lasso for all the settings which we considered. Not only in the simulated environment, the LINEX regression approach performs well with real data as well. Proceeding chapter will further summarize the results of all the chapters followed by a discussion.

Chapter 5

Discussion and Future Work

In this chapter, we present the final remarks of the thesis followed by conclusions and discussion. Furthermore, we present some further research directions as extensions of this thesis.

5.1 Conclusion and Discussion

The main objective of this thesis was to come up with new shrinkage methods which perform asymmetric shrinkage on coefficients of multiple linear regression model. Also, we wanted the user to have some control over the amount of shrinkage on each coefficient. In the first part of the thesis, we presented the literature on most widely used shrinkage methods, and also we presented some of the literature on shrinkage approaches which does asymmetric shrinkage on regression coefficients such as the adaptive lasso and generalized ridge regression. For the two predictor case, we visually illustrated the nature of the penalty region defined by each shrinkage approach and how the change in the penalty affects the corresponding coefficient estimates. Furthermore, we did a comparison between the shrinkage methods with a real data example to understand the nature of the estimated coefficients and, we compared the prediction ability between models.

In the second part of the thesis, in Chapter 3, we developed the quadratic garrote method suggested by Breiman (1995), and we further extended the idea to obtain the generalize quadratic garrote which is more flexible in the sense that the user can control the amount of shrinkage on each coefficient estimate. We derived a closed form solution for the quadratic garrote problem and studied the theoretical properties of the suggested estimator such as variance, expectation and bias. In addition, through simulation studies under different settings and with an example, we showed that the quadratic garrote is a worthy substitute for ridge regression. Furthermore, with the Boston housing dataset, we illustrated

how to use the generalized ridge regression with predefined amount of shrinkage on each coefficient based on one's experience or prior knowledge.

As the final part of the thesis, we explored the possibility of using different loss functions as the penalty in the non-negative garrote. Then, we developed the LINEX regression approach, using the LINEX loss function in place of the penalty term of the non-negative garrote. Since a closed form solution for the LINEX regression problem cannot be derived, we used a numerical optimization technique to obtain the LINEX regression coefficients estimates. We performed a simulation study under different settings and showed that LINEX regression also can be used to apply asymmetric shrinkage on regression coefficients while maintaining a more or less prediction accuracy than ridge regression.

More specifically, in high dimensional setting where the majority of population regression coefficients are small in size and few of the coefficients are considerably larger than the rest, we saw that the both quadratic garrote and LINEX regression are capable of shrinking the smaller coefficients while keeping the larger coefficients almost unchanged. Another most important observation is, in the aforementioned setting, both of the suggested methods showed much lower prediction errors compared to the ridge regression or OLS models.

Even though the suggested shrinkage methods meet our objectives, we can point out some limitations. First, since we developed the suggested methods following the idea behind the non-negative garrote, the regression coefficient estimates of each method depend on OLS estimates. Hence, the suggested methods fail when OLS estimates are infeasible. Also, as we mentioned earlier, quadratic garrote and LINEX regression do not perform subset selection. Another concern with the generalized quadratic garrote is it's subjectiveness. As we saw, in generalized quadratic garrote, the user has the ability to control the amount of shrinkage on each coefficient by defining the appropriate shrinking factors. This might incorporate some additional subjectiveness to the coefficient estimates.

Unlike ridge regression or the quadratic garrote, LINEX regression was defined with two tuning parameters namely λ and α. In this thesis, using a similar idea to the elastic net, we first fixed α and then λ was determined by cross-validation method. However, in LINEX regression, α can take any positive real value. Hence, it would be difficult for someone to decide a fixed value for α. Also, the best λ which minimizes the prediction error, changes dramatically with the choice of α. Hence, the use of the two dimensional cross-validation over a combination of α and λ might not be practical. Furthermore, the estimated coefficient at the best λ for a small value of α resulted in unnecessary shrinkage on coefficients. On the other hand, using a large α resulted in high prediction errors and the estimated model was very similar to the OLS model. However, according to the simulation studies, we

saw that using a α between 0.5 to 1, we can maintain some trade-off between the prediction error and the amount of shrinkage.

5.2 Future Work

In this thesis, all the penalty terms of the shrinkage methods that we developed were convex. However, by using a convex penalty, we cannot expect the subset selection property. Hence, one can use concave functions such as exponential loss function as the penalty terms with the non-negative garrote concept in order to obtain a shrinkage method which is capable of producing sparse models.

Another direction to further study is to use the Bayesian approach to derive the regression coefficients of each method. With the Bayesian approach, one can easily incorporate the penalty term as the prior knowledge. Assume the multiple linear regression model $y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, i = 1, 2, ..., n$, where $\epsilon_i \sim$ iid $\mathrm{N}(0, \sigma^2)$. Let $f(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta})$ be the likelihood function given by

$$\begin{aligned} f(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}) &= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2\right\}, \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\right\}. \end{aligned} \tag{5.1}$$

In the Bayesian aspect of the regression analysis, we assume that the population coefficient vector, $\boldsymbol{\beta}$ to have a prior distribution $p(\boldsymbol{\beta})$. Assuming $\boldsymbol{X}$ to be fixed, the posterior distribution is obtained as

$$p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) \propto f(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta})p(\boldsymbol{\beta}). \tag{5.2}$$

Then, the mode of the posterior is given by

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{argmax}}\ \{p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y})\} &= \underset{\boldsymbol{\beta}}{\text{argmax}}\ \{f(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta})p(\boldsymbol{\beta})\}, \\ &= \underset{\boldsymbol{\beta}}{\text{argmax}}\ \left\{\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\right] p(\boldsymbol{\beta})\right\}, \\ &= \underset{\boldsymbol{\beta}}{\text{argmax}}\ \left\{-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \log[p(\boldsymbol{\beta})]\right\}, \\ &= \underset{\boldsymbol{\beta}}{\text{argmin}}\ \left\{\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) - \log[p(\boldsymbol{\beta})]\right\}. \end{aligned}$$

Let $p(\boldsymbol{\beta}) = \prod_{j=1}^{p} g(\beta_j)$. Then the mode of the posterior can be obtained as the minimizer of

$$\mathrm{Q}(\boldsymbol{\beta}) = \frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \sum_{j=1}^{p} -\log[g(\beta_j)]. \tag{5.3}$$

When $g(\beta_j) \sim \mathrm{N}(0, \sigma^2/\lambda)$, the minimization problem in (5.3) can be simplified into

$$\mathrm{Q}(\boldsymbol{\beta}) = \frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \frac{\lambda}{2\sigma^2}\sum_{j=1}^{p}\beta_j^2,$$

which is nothing but the ridge regression problem. Similarly, when $g(\beta_j)$ has a Laplace distribution with the scale parameter $1/\lambda$, (5.3) becomes the lasso problem. Also, we can show that, when $g(\beta_j) \sim \mathrm{N}(0, \sigma^2/\lambda d_j^2)$, the minimization problem in (5.3) becomes the generalized quadratic garrote problem. In this case, we can conveniently define d_j^2's based on the prior knowledge or experience. Also, one can use different distributions as the prior to develop different types of shrinkage methods such as the methods which we introduced in this thesis. The other advantage of using Bayesian approach is the simplicity of estimating regression coefficients. Once we obtained the appropriate prior and then the posterior, the parameters are usually obtained as the mode of the posterior distribution. Obtaining the mode of the posterior distribution is usually easier than the direct optimization of (5.3). If the posterior distribution is a known distribution, we can obtain the mode theoretically. If the posterior distribution is unknown, we still can use Markov chain Monte Carlo (MCMC) method or Simulated Annealing (SA) method to approximately estimate the coefficients.

Bibliography

Akdeniz, F. (2004). New biased estimators under the linex loss function. *Statistical Papers 45*(2), 175–190. (Cited on page 65.)

Boer, P. and C. M. Hafner (2005). Ridge regression revisited. *Statistica Neerlandica 59*(4), 498–505. (Cited on pages 13 and 14.)

Breheny, P. (2015). The group exponential lasso for bi-level variable selection. *Biometrics 71*(3), 731–740. (Cited on pages 5 and 65.)

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 373–384. (Cited on pages 4, 5, 6, 21, 22, 24, 26, 33, 34, 45, 47, 48 and 97.)

Byrd, R. H., P. Lu, J. Nocedal, and C. Zhu (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing 16*(5), 1190–1208. (Cited on page 68.)

Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004). Least angle regression. *The Annals of statistics 32*(2), 407–499. (Cited on page 17.)

Hager, W. W. (1989, June). Updating the inverse of a matrix. *SIAM Rev. 31*(2), 221–239. (Cited on page 39.)

Harrison, D. and D. L. Rubinfeld (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management 5*(1), 81–102. (Cited on page 27.)

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York. (Cited on pages 26 and 50.)

Hemmerle, W. J. (1975). An explicit solution for generalized ridge regression. *Technometrics 17*(3), 309–314. (Cited on page 14.)

Henderson, H. V. and S. R. Searle (1981). On deriving the inverse of a sum of matrices. *Siam Review 23*(1), 53–60. (Cited on page 39.)

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67. (Cited on pages 5, 8, 10, 12, 14 and 36.)

Horel, A. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress 58*, 54–59. (Cited on pages 2, 4 and 8.)

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer. (Cited on pages 4, 25, 26 and 29.)

Lawson, C. L. and R. J. Hanson (1974). *Solving least squares problems*. (Cited on page 24.)

Ohtani, K. (1995). Generalized ridge regression estimators under the linex loss function. *Statistical Papers 36*(1), 99–110. (Cited on page 65.)

Rao, C., H. Toutenburg, and H. Shalabh (2008). C.: Linear models and generalizations: Least squares and alternatives. (Cited on page 2.)

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics 22*(2), 231–245. (Cited on page 18.)

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. (Cited on pages 4, 14, 17, 21 and 50.)

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(1), 91–108. (Cited on page 18.)

Varian, H. R. (1975). A bayesian approach to real estate assessment. *Studies in Bayesian Econometric and Statistics in honor of Leonard J. Savage*, 195–208. (Cited on page 65.)

Wu, T. T. and K. Lange (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 224–244. (Cited on page 17.)

Yoo, W., R. Mayberry, S. Bae, K. Singh, Q. P. He, and J. W. Lillard Jr (2014). A study of effects of multicollinearity in the multivariable analysis. *International journal of applied science and technology 4*(5), 9. (Cited on page 2.)

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67. (Cited on page 18.)

Zhang, K., F. Yin, and S. Xiong (2014). Comparisons of penalized least squares methods by simulations. *arXiv preprint arXiv:1405.1796*. (Cited on page 53.)

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association 101*(476), 1418–1429. (Cited on pages 5, 18, 19 and 25.)

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320. (Cited on pages 19 and 20.)