Variable Selection in Regression Analysis

Author: Mohamed Aidiel Haikal Mat Ziat
Student ID: 10496505
Supervisor: Dr. Christiana Charalambous
Module Code: MATH30000

2021/2022

Abstract

An important issue arising in regression analysis is how to choose which covariates to include in the modelling process. We would prefer to fit a model that is not only best at explaining the response variable but also computationally efficient. Hence, we would try eliminating least promising variables and retain the significant ones through the use of variable selection techniques. These techniques are fundamental in tackling this issue especially in high- dimensional statistical modelling.

In this report, we will investigate some of the most popular variable selection techniques, study the theory behind them and compare their performance through simulation studies and real data analysis. This project will include reviewing traditional variable selection methods such as stepwise regression and subset selection in addition to shrinkage-based methods such as ridge, LASSO and SCAD.

Contents

Chapter 1

Introduction

1.1 Motivation

This report is motivated by the importance of variable selection in general regression settings, especially when dealing with *high-dimensional* data. Dimension in this context refers to the number of parameters, p. Data sets containing more features than observations, denoted as $p \gg n$ are referred to as *high-dimensional*.

In this report, we limit ourselves to study the less extreme case where we have large number of predictors but still less than the number of observations ($p < n$). The additional signal features which are not related to the response will introduce noise to the fitted model, leading to deterioration in prediction performance. To compensate this issue, statisticians have employed wide range of variable selection techniques which can be classified into three major classes (James et al., 2014):

- *Subset selection.* In this approach we select a subset of the p predictors which are found to be related to the response based on some measures. The selected predictors are then used to fit a new model using the ordinary least squares.
- *Shrinkage.* This approach fits all p predictors but the estimated coefficients are shrunken towards zero due to the penalty imposed on the regression model. Shrinkage methods perform variable selection.
- *Dimension reduction.* The p predictors are projected into M-dimensional subspace, with $M < p$. These M projections are then used as predictors to fit a linear regression model by least squares.

We will be focusing on the first two classes which are subset selection and shrinkage-based methods.

1.2 Background theory

We begin our discussion by introducing the *multiple linear regression* model. Suppose we have a set of data $(\mathbf{X}_i, y_i)$ for $i = 1, 2, ..., n$, where $\mathbf{X}_i = (X_{i1}, X_{i2}, ..., X_{ip})$ is the $n \times p$ matrix of predictors with corresponding responses, y_i. Then, the multiple linear regression can be modelled as

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i, \tag{1.1}$$

with intercept β_0, regression coefficients β_j and error ϵ_i. The regression coefficients, β_j measures the degree of change in y_i for every unit of change in X_i. The error term, ε_i measures the discrepancies between the response y_i and its mean, $\mu_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j$. If $p = 1$, the model is called *simple linear regression* formulated as follows,

$$y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Writing model (1.1) in matrix form gives us

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & X_{21} & \dots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & \dots & X_{np} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and,} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Given the system of equations, our aim in regression analysis is to find the estimate of regression coefficients, $\hat{\boldsymbol{\beta}}$ which "*best*" fit the equations. The basic idea is to use ordinary least squares (OLS). We define the sum of squared errors as

$$S(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{1.2}$$

The least squares estimators are obtained by minimising (1.2) i.e.,

$$\hat{\boldsymbol{\beta}}^o = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \tag{1.3}$$

The usual way of minimising (1.2) is to differentiate the equation with respect to $\boldsymbol{\beta}$. Thus, we have

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta},$$
$$\frac{\partial^2 S}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} = 2\mathbf{X}^T\mathbf{X}.$$

Assuming $\mathbf{X}$ has full rank, this implies $\mathbf{X}^T\mathbf{X}$ is positive definite. Hence, (1.2) is strictly convex function and the solution, $\hat{\boldsymbol{\beta}}^o$ is a unique global minimum. Setting the first derivative to zero leads to a system of linear equations known as the normal equations,

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}.$$

Consequently, we get the unique least squares solution for equation (1.3) as

$$\hat{\boldsymbol{\beta}}^o = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1.4}$$

Note that the solution cannot be uniquely determined if our assumption $\mathbf{X}$ is full rank, does not hold. If $\text{rank}(\mathbf{X}) < p+1$, then $\mathbf{X}^T\mathbf{X}$ is rank deficient and is singular. As a result, the least square coefficients $\hat{\boldsymbol{\beta}}$ is not unique. This situation would occur if perfect correlation exists between some of the predictors, (e.g., $\mathbf{X}_3 = 2\mathbf{X}_1$). Moreover, rank deficiencies can also occur in high-dimensional data.

Given the least square coefficients estimates (1.4), we have the fitted model given by

$$\hat{\mathbf{y}} = \hat{\mu}(\mathbf{X}) = \mathbf{X}\hat{\boldsymbol{\beta}}^o = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y},$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is called the "hat" matrix. The least squares residuals are then defined as

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^o = (\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y} = (\mathbf{I} - \mathbf{H})\mathbf{y} = \mathbf{M}\mathbf{y},$$

where $\mathbf{M} = \mathbf{I} - \mathbf{H}$. Then, the residual sum of squares (RSS) is given by

$$\text{RSS}(\hat{\boldsymbol{\beta}}^o) = \mathbf{e}^T\mathbf{e} = \mathbf{y}^T\mathbf{M}\mathbf{y} = \mathbf{y}^T(\mathbf{I} - \mathbf{H})\mathbf{y} = \mathbf{y}^T\mathbf{y} - (\hat{\boldsymbol{\beta}}^o)^T\mathbf{X}^T\mathbf{y} \tag{1.5}$$

where the first term is called total sum of squares, TSS and the second term is the explained sum of squares, ESS or in other words error due to the regression.

We can define the coefficient of determination, R^2 as

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}. \tag{1.6}$$

The R^2 is a statistical measure that represents the proportion of variance in the response variable explained by the covariates in the regression model. A value close to 1 implies most of the variation in the response variable are captured by the regression model, while a value close to 0 implies the opposite.

In order to fit a linear regression model, the following assumptions are made:

1. Model is linear in terms of the beta parameters.
2. The response y_i are uncorrelated and have a constant variance σ^2.
3. The predictors x_i are fixed (non-random).
4. The errors are homoskedastic which means the variance of errors are constant.
5. The errors are independent.
6. The mean for these errors is zero.
7. Optionally, the errors follow multivariate normal distribution (MVN) i.e., $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$

Following the assumptions, we can obtain some sampling properties of the least square coefficients estimates $\hat{\boldsymbol{\beta}}^o$. The variance-covariance matrix of the ordinary least squares parameter estimates is derived from (1.4) as follows,

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\beta}}^o) &= \text{Var}\big((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\big) \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\text{Var}(\mathbf{y})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\sigma^2\mathbf{I})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\
&= \sigma^2\mathbf{I}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}
\end{aligned}$$

From the above, we still have to estimate the population variance σ^2. The estimate can be derived by taking the expectation of (1.5) and is given by

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\text{RSS}(\hat{\boldsymbol{\beta}}^o) \tag{1.7}$$

The following points summarize the useful properties of least square estimates $\hat{\boldsymbol{\beta}}^o$:

1. Given $\mathbf{X}$ is full rank, $\hat{\boldsymbol{\beta}}^o$ is unique and linear in the response vector $\mathbf{y}$.
2. $\hat{\boldsymbol{\beta}}^o$ is unbiased, $E(\hat{\boldsymbol{\beta}}^o) = \boldsymbol{\beta}$.
3. $\text{Var}(\hat{\boldsymbol{\beta}}^o) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$
4. The least square estimates are scale invariant which means regardless of how the jth predictor is scaled, the product $X_j\beta_j$ will remain the same.

Based on the above, the least squares estimates have the best prediction accuracy among all linear and unbiased estimates. In the case of $n \gg p$, the variance of OLS will be small and have a decent prediction accuracy. Unfortunately, when we have large p, which is close, but less than n, the least squares estimates tend to be highly variable resulting in poor prediction performance. Also as mentioned previously, least square is not even a viable option in high-dimensional settings when $p \gg n$ since it no longer provides a unique coefficient estimates. Removing additional predictors which are less related to the response variable will produce a sparse model which is easier to interpret, and significantly help in improving the prediction accuracy. However, it is highly unlikely that OLS will set any of the parameter estimates to zero.

For those reasons, variable selection methods have been introduced to solve such problems and attain desirable results. The purpose of this report is to review some of the traditional methods such as stepwise regression and also shrinkage-based methods which includes the *least absolute shrinkage and selection operator* (LASSO), *smoothly clipped absolute deviation*(SCAD), elastic net and ridge regression. In addition to that, we will also discuss some extension to LASSO such as adaptive LASSO and group LASSO. We will be comparing and contrasting the performance of these methods in simulation studies and also application to real data.

This report is organized as follows. Chapter 2 examines the traditional methods available for variable selection, followed by the modern shrinkage methods. In Chapter 3, we discuss some of the available methods for model selection and also optimizing the tuning parameters. Comprehensive simulation studies are presented in Chapter 4. Chapter 5 presents the application of some selected methods on real-life data and concluding remarks are given in Chapter 6.

Chapter 2

Methods for Variable Selection

This chapter provides an overview of the methods available for variable selection. Section 2.1 looks at the traditional methods available to overcome the problems in least squares. Some of the well-known methods include best-subset selection 2.1.1, and stepwise regression 2.1.2, in particular forward selection, backward elimination and the full stepwise regression. Shrinkage methods are discussed in 2.2.

2.1 Subset selection

Subset selection methods attempt to find the correct subset of k variables from all the available p predictors for model specification. Once the subset has been identified, we fit a model using least squares to estimate the coefficients retained in the model.

We can interpret the problem of selecting $k < p$ predictors as finding the $p - k$ predictors which are not related to the response variable and setting the corresponding beta coefficients to zero. Suppose the first k predictors are selected. Omitting the intercept, we have the least squares model 1.1 reduced to

$$y_i = \sum_{j=1}^{k} X_{ij}\beta_j + \varepsilon_i$$

for $i = 1, 2, ..., n$.

2.1.1 Best-subset selection

To execute best-subset selection, we fit a separate least squares regression for all subsets of predictors of size k where $k \in \{1, 2, .., p\}$. These subsets contain every possible combination of predictors, p. We aim to find the subset that yields the smallest residual sum of squares (1.5). As summarized in James et al. (2014), the algorithm to perform best subset selection is as follows,

Algorithm 2.1.1 (Best-subset selection)**.**

1. *Fit the null model, $\mathcal{M}_0$ which contains only the intercept.*
2. *For $k = 1, 2, ..., p$:*
   (a) *Fit all $\binom{p}{k}$ possible models of size k.*
   (b) *Select model with smallest RSS (consequently, largest R^2) from the $\binom{p}{k}$ models and define it as model $\mathcal{M}_k$.*
3. *Select a single best model among the $\mathcal{M}_0, ..., \mathcal{M}_p$ based on some selection criteria.*

In Algorithm 2.1.1, Step 2 reduces the problem of selecting the best model from 2^p possible models to $p + 1$ possible models. From these $p + 1$ models, we have to select a single subset containing k predictors that gives the *best* model. The best of these models is selected using one of the information criteria or cross-validation (CV) discussed in Chapter 3.

Efficient algorithms available to perform the exhaustive for the best subsets of the variables include the *leaps and bounds* (Furnival and Wilson, 1974) and the *branch and bound* which is used in `leaps` package in R (Thomas Lumley, 2020).

2.1.2 Stepwise regression

Stepwise regression (Breaux, 1967) seek a good path and explore a more restricted set of models instead of searching through all possible subsets.

Forward selection

The first approach of stepwise regression is *forward selection*. We start with the null model, containing the constant term if required, otherwise the null model contains no terms at all. Next, consider adding predictors, one-at-a-time, that most improves the fit. Note that once a term is added, it is retained in all subsequent subsets of predictors. The step is repeated until no significant predictor to be added into the model that further improves the fit. In general, the algorithm for forward selection is given below.

Algorithm 2.1.2 (Forward selection)**.**

1. *Let $\mathcal{M}_0$ denote the null model, which contains no predictors.*
2. *For $k = 0, 1, ..., p-1$:*
   (a) *Fit all $p-k$ subsets of size $k+1$ which add one variable to subset $\mathcal{M}_k$.*
   (b) *Select model with smallest RSS from the $p-k$ models and define it $\mathcal{M}_{k+1}$.*
3. *Select a single best model from among $\mathcal{M}_0, ..., \mathcal{M}_p$ using some selection criteria.*

Forward selection involves fitting a single null model and $p-k$ models throughout the iterations for $k = 0, ..., p-1$. This amounts to a total of $1 + \sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models which is significantly lower, for large p, than best subset selection with 2^p models. Along with the computational advantage it provides, forward selection can also be applied even in a high-dimensional setting, when $p \gg n$.

Backward elimination

On the contrary, a *backward elimination* starts with the full least squares model containing all p predictors. Least useful predictors are then iteratively removed, one-at-a-time. The full details of the algorithm are shown below.

Algorithm 2.1.3 (Backward elimination)**.**

1. *Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.*
2. *For $k = p, p-1, ..., 1$:*
   (a) *Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.*
   (b) *Select the subset of size $k-1$ which has the smallest RSS and call it $\mathcal{M}_{k-1}$.*
3. *Select a single best model from among $\mathcal{M}_0, ..., \mathcal{M}_p$ using some selection criteria.*

As backward elimination is intuitively the reversed of forward selection, the total number of models to be considered is the same which is $1 + p(p+1)/2$. Hence, backward elimination shares the same computational advantage as forward selection. A key difference between the two approaches is backward selection requires the number of samples n to be larger than number of variables, p. Therefore, it is not useful in a high-dimensional setting.

Full stepwise regression

Full stepwise regression is the hybrid version of forward selection and backward elimination. We start with a minimal model and variables are added sequentially, similar to the forward selection. However, rather than simply retaining the variables in the model, this approach will also remove any variables that no longer provide an improvement to the model fit. This method mimics the best subset selection while retaining the computational advantages of forward and backward stepwise regression. The algorithm can be summarized as follows.

Algorithm 2.1.4 (Full stepwise)**.**

1. *Start with a minimal model containing the intercept if required, otherwise the model contains no terms.*
2. *Consider adding a single term which a yield a model with the smallest RSS.*
3. *Fit the model with the new term added.*
4. *Identify a term which give the smallest RSS when the term is removed from the fitted model.*
5. *Fit a new model with that term deleted.*
6. *Go back to step 2, and continue cycling through steps 2,3 and 4 until the cycle produces no change, i.e. no more terms can be added and deleted to improve the fit.*

Alternatively, Step 2-5 in the Algorithm 2.1.4 can be replaced by conducting an F-test to assess the fitted model. Thus, the algorithm can be interpreted as a sequence of hypothesis tests which we test if the model containing the added variable is the true model. This approach is discussed in detail in Draper and Smith (1966). At each iteration in Step 2, the variable which produces a model with a larger F-statistic than a specified value, say F_{in} will be included in the model. After adding a term, the variable with the lowest F-statistic is then compared to another specified value, say F_{out}. The variable is removed if its F-statistic is lower than F_{out}. Terminate the algorithm when no terms can be added or removed from the model. The value of F_{in} and F_{out} can be set to be equal to the critical values of F-distribution corresponding to a suitable significance level, α. This approach is also applicable for forward selection and backward elimination with some adjustments.

From this section, we can see that in cases where parsimonious model is desired, subset selection methods help to overcome the problem with least squares as they provide a reduced model which eases interpretation and prevents overfitting. Moreover, forward selection method, as discussed above, is also applicable in high-dimensional settings, making it more useful than the least squares.

However, selecting a subset of variables introduces bias to the estimated beta coefficients. The bias tends to be very large especially in methods like best-subset selection due to the large number of models need to be considered. Apart from that, the result produced by these methods are not consistent due to

its tendency to select a local minimum instead of the global minimum which we aim to achieve. Moreover, the inconsistency in estimation also arise due to their discrete nature in selecting variables. At each step of the algorithm, variables are either retained or removed (corresponding coefficient set to zero). This results in extremely high variability in the estimates depending on which covariates are excluded from the model. Many papers including Hurvich and Tsai (1989), Steyerberg (1999), and Flom and Cassell (2007) have proved biased estimation and inconsistent selection arising from subset selection methods. A criticism from Doornik (2009) points out the lack of search in subset selection methods. That is, the algorithm never considers to test previously removed variables after each step. The full stepwise regression discussed above is restricted to testing the previous step only. Thus, subset selection methods might behave poorly in variable selection.

2.2 Shrinkage methods

As opposed to subset selection, shrinkage methods stem on the idea of applying penalty on parameters which constraints the coefficient estimates, or equivalently *shrinks* the estimates towards zero. This encourages sparsity in the fitted model hence can be better interpreted. As shrinkage methods are continuous, they do not suffer from high variability as seen in subset selection.

Desboulets (2018) classifies penalty-based methods into two categories: norm penalties and also concave penalties. The norm penalty takes the form:

$$\underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_\gamma^2,$$

where γ varies depending on the methods. For example, $\gamma = 1$ corresponds to LASSO and $\gamma = 2$ is used in ridge regression. On the other hand, the concave penalties provide a general framework:

$$\underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + p_\lambda(\boldsymbol{\beta}),$$

where the choice of penalty, $p_\lambda(\boldsymbol{\beta})$ varies according to each method.

Following the notion, we can classify the methods discussed in this report as follows.

Model	Type of penalty	
	Norm	Concave
Linear	Ridge LASSO BRidge Adaptive LASSO	SCAD Non-Negative Garotte
Group	Elastic Net Group LASSO	

Table 2.1: Classification of penalized methods

One of the properties that we are interested in is the convexity of each method. Convexity allows a particular method to have a certain advantages which we will discuss later. For completeness, we will include the definition of convex and strictly convex function.

Definition 2.2.1. *A set $S \subseteq \mathbb{R}^n$ is convex if for all $\mathbf{x}, \mathbf{y} \in C$ and $\lambda \in [0,1]$, the line $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in S$.*

Definition 2.2.2. *Consider a set $S \subseteq \mathbb{R}^n$. Then a function $f : S \to \mathbb{R}$ is called convex if S is convex and for all $\mathbf{x}, \mathbf{y} \in S$ and $\lambda \in [0,1]$,*

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}).$$

The function f is strictly convex if

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}).$$

A function f is concave if $-f$ is convex.

In this section, we first present the two best-known shrinkage methods which are ridge regression 2.2.1 and the LASSO 2.2.2. We will also discuss BRidge 2.2.3 which provides a general framework of ridge and LASSO, followed by non-negative garrote (NNG) 2.2.4 which serves as the motivation of LASSO. We then discuss some improved shrinkage methods namely SCAD 2.2.5, adaptive LASSO 2.2.6, elastic net 2.2.7 and group LASSO 2.2.8.

2.2.1 Ridge regression

The idea of ridge regression was proposed by Hoerl and Kennard (2000). Ridge regression improves the accuracy of the OLS estimates by minimizing the sum of squared error (1.2) bounded on L_2-norm of the coefficients. As the constraint is placed on the size of parameters, it is crucial that all predictors are on the same scale to allow equal consideration. This implies that ridge regression is not scale invariant which means parameter estimates can change drastically when the scale of predictors varies. Therefore, we assume that the predictors, X_{ij} are standardized such that $\sum_i X_{ij}/n = 0$ and $\sum_i X_{ij}^2/n = 1$. The ridge estimates $\hat{\beta}^R$ can be written as

$$\hat{\beta}^R = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \Big(y_i - \sum_{j=1}^{p} \beta_j X_{ij} \Big)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{2.1}$$

where $\lambda \geq 0$ is a tuning parameter which controls the magnitude of shrinkage (Hastie, 2008). Equivalently,

$$\hat{\beta}^R = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2, \qquad \text{subject to} \sum_{j=1}^{p} \beta_j^2 \leq t. \tag{2.2}$$

There is a one-to-one correspondence between λ in (2.1) and t in (2.2). When $\lambda = 0$, the penalty term has no effect. Thus, the ridge regression is equal to the least squares in this case. As we increase λ, i.e., $\lambda \to \infty$, more penalty is applied

and the parameter estimates, $\hat{\beta}^R \to 0$. Unlike least squares which produces a single set of coefficient estimates, ridge regression generates a set of parameter estimates for each value of λ, denoted as $\hat{\beta}^R(\lambda)$. As such, it is crucial to select a good value for λ. We defer this discussion to Chapter 3, where we discuss cross-validation. We can plot the path of each parameter estimates as we increase the value λ. The visualisation of coefficient paths is available in Chapter 5. Notice that we omit the intercept term, β_0 in both (2.1) and (2.2) as we can assume, without loss of generality, that the solution for β_0 is $\hat{\beta}_0 = \bar{y} = 0$ for all t.

We can write the criterion of (2.1) in matrix form as

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

and the ridge penalty is given as

$$p_R(\lambda) = \lambda\beta^T\beta. \tag{2.4}$$

Note that this function is twice differentiable and is strictly convex when $\lambda > 0$ for all $\boldsymbol{\beta}$. Since the penalty function is differentiable, we can derive the ridge solution explicitly. Setting the partial derivatives of (2.3) to zero, we have

$$\begin{aligned}
& \frac{\partial}{\partial\beta}\Big\{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta\Big\} = 0 \\
\Leftrightarrow \quad & \frac{\partial}{\partial\beta}\Big\{\mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta + \lambda\beta^T\beta\Big\} = 0 \\
\Leftrightarrow \quad & -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\beta + 2\lambda\beta = 0 \\
\Leftrightarrow \quad & \mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{X}\beta + \lambda\beta \\
\Leftrightarrow \quad & \mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\beta.
\end{aligned}$$

Therefore, the ridge estimator is

$$\hat{\beta}^R = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.5}$$

where $\mathbf{I}$ is the $p \times p$ identity matrix.

The variance of the ridge estimates can easily be derived in a similar way to the variance of the least square estimates, and is given as

$$\mathrm{Var}(\hat{\beta}^R) = \sigma^2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}.$$

It is less trivial to show that ridge estimates have lower variance compared to the least squares in this setting. Thus, we now consider *orthonormal design case*, where the *information matrix*, $\mathbf{X}^T\mathbf{X} = \mathbf{I}$. For a certain j, the ridge estimates defined in terms of the least squares estimates are given as

$$\hat{\beta}_j^R = \frac{1}{1+\lambda}\hat{\beta}_j^o, \tag{2.6}$$

where $\hat{\beta}_j^o$ is the full least square estimates. From equation 2.6 above, it can easily be shown that the variance for ridge estimates is certainly lower than least squares:

$$\mathrm{Var}(\hat{\beta}_j^R) = \left(\frac{1}{1+\lambda}\right)^2 \mathrm{Var}(\hat{\beta}_j^o).$$

This result is expected as penalizing the coefficients introduces a small bias to the fitted model, in exchange for a reduction in variance.

We can acquire some insight about the behaviour of this solution by plotting (2.6) as shown in Figure 2.1(b). Based on the plot, we can see that ridge regression scales the coefficients by a constant factor but none are set to zero. Even if we increase the value of λ, the optimal slope $\hat{\beta}_j^R$ will approach 0 but never actually equals to 0. As a result, ridge regression will always keep all the predictors in the model which does not give a sparse model that is easy to interpret. Hence, ridge regression is only capable of performing coefficients shrinkage but not variable selection.

2.2.2 LASSO

The two previously discussed methods for improving the least squares estimates, subset selection and ridge regression both have their own drawbacks. Ridge regression able to produce stable and accurate result since it is a continuous process which select variables and estimates coefficients simultaneously. However, its inability to set any coefficients to zero makes it less useful in producing a model that is easy to interpret. On the other hand, subset selection able to produce a sparse model which easy to interpret but can be extremely variable due to its nature being discrete process.

Thus, the *'least absolute shrinkage and selection operator'* (LASSO) was proposed by Tibshirani (1996) to overcome both shortcoming of ridge regression and subset selection. Instead of using L_2-norm, LASSO imposes L_1-norm on the regression coefficients. As a result, it is able to shrink some coefficients while others are set to 0. Hence, we obtain a sparse model without compromising the prediction accuracy.

The only difference between LASSO and ridge regression is the penalty term imposed on the sum of squared errors (1.2). The Lagrange form of LASSO estimates is defined as

$$\hat{\beta}^L = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.7}$$

or its equivalent constrained form,

$$\hat{\beta}^L = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 \right\} \quad \text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t, \tag{2.8}$$

where λ and t are the tuning parameter, often chosen by model selection procedure such as cross-validation (refer Chapter 3). Similarly, there is a one-to-one correspondence between λ and the bound t. Notice that we also omit the intercept term in LASSO for the same reason mentioned in Section 2.2.1.

The LASSO penalty takes the form

$$p_L(\lambda) = \lambda \sum_{j=1}^{p} |\beta_j|,$$

which is convex and is always positive. Thus, the minimum is achieved when β_j are all close to zero. This implies that LASSO also shrinks parameter estimates

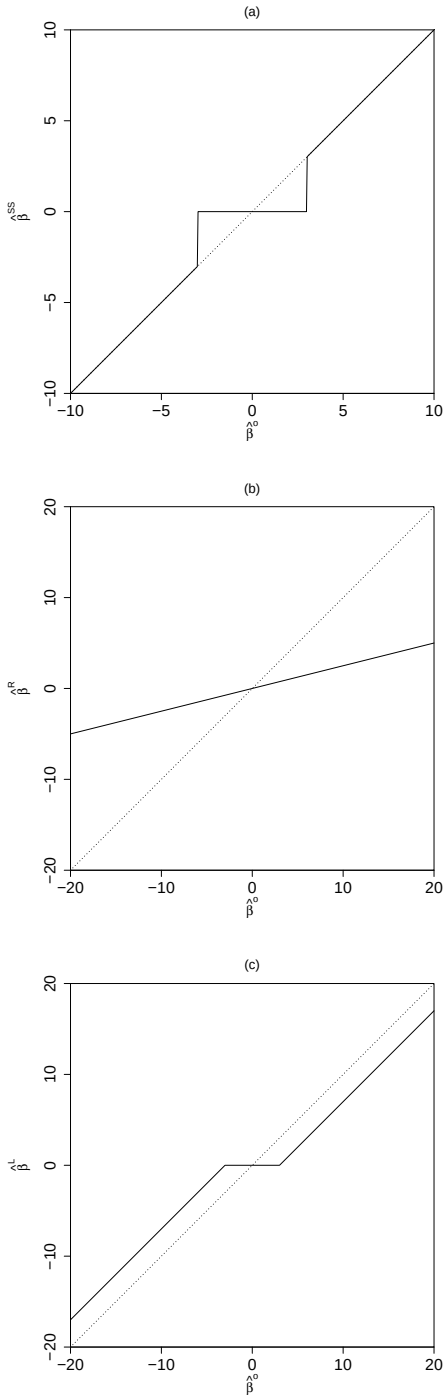


Figure 2.1: Plot of the thresholding functions for (a) subset selection (hard thresholding), (b) ridge regression, and (b) LASSO with $\lambda = 3$. The 45° dotted lines represents the full least square estimates.

to zero. However, the LASSO penalty is non-differentiable at $\beta_j = 0$, making it has the effect of setting some parameter estimates exactly to zero. Therefore, LASSO is able to perform variable selection as well as coefficient shrinkage.

In the orthonormal design case, the closed form solution to equation (2.8) can easily be derived, in a similar way as the ridge estimates, to be

$$\hat{\beta}_j^L = \text{sign}(\hat{\beta}_j^o)(|\hat{\beta}_j^o| - \lambda)_+ \tag{2.9}$$

where $sign(\cdot)$ denotes the sign of its argument (± 1) or 0 if its argument is 0, and x_+ denotes the positive part of x which is x if $x > 0$ and 0 otherwise. The solution can be written as

$$\mathcal{S}(\hat{\beta}_j^o, \lambda) = \begin{cases} \hat{\beta}_j^o - \lambda & \text{if } \hat{\beta}_j^o > \lambda, \\ 0 & \text{if } |\hat{\beta}_j^o| \leq \lambda, \\ \hat{\beta}_j^o - \lambda & \text{if } \hat{\beta}_j^o < -\lambda. \end{cases} \tag{2.10}$$

This is referred to as 'soft thresholding rule' (Donoho and Johnstone, 1994), where we reduce $\hat{\beta}_j^o$ by a fixed constant λ, without letting it go negative.

From our discussion of ridge regression and LASSO, we found that the parameter estimates for both methods can be expressed in terms of the least squares estimates under the orthogonal design case. An interesting fact is the same approach is also applicable for subset selection methods discussed previously. The problem of finding a subset of size $k < p$ parameter estimates can be restated as

$$\hat{\beta}^{SS} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2, \qquad \text{subject to} \quad \sum_{j=1}^{p} \delta(\beta_j \neq 0) \leq k.$$

where $\delta(\beta_j \neq 0)$ is the indicator function which equals to 1 if $\beta_j \neq 0$ and 0 if $\beta_j = 0$. The problem is equivalent to the penalized regression

$$\hat{\beta}^{SS} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \frac{\lambda^2}{2} \sum_{j=1}^{p} \|\beta_j\|_0,$$

where the penalty function is the so-called L_0-norm (Donoho and Elad, n.d.). Note that the L_0-norm, $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^{p} \delta(\beta_j \neq 0)$ is not an actual norm but simply the number of non-zero components of vector $\boldsymbol{\beta}$.

Given the discrete nature of the L_0-norm, we can minimize the objective function above in the orthogonal design as

$$H_{SS}(\beta_j) = \frac{1}{2}(\hat{\beta}_j^o - \beta_j) + \frac{\lambda^2}{2}\delta(\beta_j \neq 0).$$

Now, if $\beta_j = 0$, then $H_{SS}(0) = (\hat{\beta}_j^o)^2/2$. On the other hand, $\beta_j \neq 0$ would imply $H_{SS}(\beta_j) = (\hat{\beta}_j^o - \beta_j)^2 + \lambda^2/2$ and the corresponding minimum is obtained when $\beta_j = \hat{\beta}_j^o$ with $H_{SS}(\hat{\beta}_j^o) = \lambda^2/2$. Hence, $\hat{\beta}_j^o$ is a solution if

$$H_{SS}(\hat{\beta}_j^o) < H_{SS}(0) \quad \Leftrightarrow \quad \lambda^2/2 < (\hat{\beta}_j^o)^2/2 \quad \Leftrightarrow \quad |\hat{\beta}_j^o| > \lambda,$$

and otherwise the solution is 0. Therefore, we have

$$H(\hat{\beta}_j^o, \lambda) = \begin{cases} \hat{\beta}_j^o & \text{if } |\hat{\beta}_j^o| > \lambda, \\ 0 & \text{if } |\hat{\beta}_j^o| \leq \lambda, \end{cases} \tag{2.11}$$

which is known as the hard thresholding rule.

We can inspect the behaviour of the solution in comparison to ridge and LASSO (soft thresholding) solutions by plotting equation 2.11 as shown in Figure 2.1(a). As we can see, subset selection does not perform any shrinkage to the parameter estimates and discretely set some coefficients to zero. On the other hand, LASSO in Figure 2.1(c), compromise between the ridge regression and subset selection by setting some parameter estimates to zero while shrinking others based on their size as discussed previously.

Computation

There are numerous approach to solve the LASSO optimization problem. Algorithms such as the shooting algorithm (Fu, 1998), the local quadratic approximation (Fan and Li, 2001), the least angle regression (LAR, Efron et al. (2004)), and coordinate descent (Wu and Lange, 2008) are well-implemented in most software packages. In the following part, we will be discussing the coordinate descent and LAR algorithm in particular as the R packages we are using; `glmnet`C and `lars`, utilizes those algorithms to compute the regularization path for LASSO and some other methods in the simulation studies (refer Chapter 4),

Cyclic Coordinate Descent

In this part, we will provide the outline of the modified version of coordinate descent called cyclic coordinate descent studied in Friedman et al. (2010b). The idea of coordinate descent is to fix the penalty parameter λ in the Lagrangian form (2.7) and successively optimize over each parameter, holding the other parameters fixed at their current values. Consider the coordinate descent step to solve (2.7). That is, suppose we have $\tilde{\beta}_k(\lambda)$ which is the current estimate for β_k at penalty parameter λ. We wish to partially optimize with respect to β_j, where $j \neq k$. We can rearrange (2.7) to isolate β_j which gives the objective function

$$R(\tilde{\beta}_k(\lambda), \beta_j) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} X_{ik}\tilde{\beta}_k(\lambda) - X_{ij}\beta_j\right)^2 + \lambda\sum_{k\neq j}|\tilde{\beta}_k(\lambda)| + \lambda|\beta_j|.$$

Note that, without loss of generality, we added a factor 1/2 for convenience of calculation. The objective is equivalent to a univariate lasso problem with partial residual $y_i - \tilde{y}_i^{(j)} = y_i - \sum_{k\neq j} X_{ik}\tilde{\beta}_k(\lambda)$ as the response variable. The objective function results in the update

$$\tilde{\beta}_j(\lambda) \longleftarrow \mathcal{S}\left(\sum_{i=1}^{n} X_{ij}(y_i - \tilde{y}_i^{(j)}), \quad \lambda\right), \tag{2.12}$$

where $\mathcal{S}(t, \lambda) = \text{sign}(t)(|t| - \lambda)_+$ is the soft thresholding operator (2.10). The first argument of the soft-thresholding operator (2.12) is the simple least-squares

coefficient of the partial residual on the standardized variable X_{ij}. In order to obtain the lasso estimate $\hat{\beta}_j^L(\lambda)$, repeat the iteration (2.12), where we cycle through each variable in turn until convergence is achieved.

This algorithm is implemented by the `glmnet` package in R. Thus, the same algorithm is applicable to compute the solutions for methods which use the R package, namely ridge regression, elastic net and adaptive LASSO. The same kind of algorithm is also applicable to many other methods such as group LASSO which have the penalty as a sum of functions of individual parameters (Friedman et al., 2010b). The cyclic modification in this algorithm increases the rate of convergence than the basic coordinate descent discussed in Wu and Lange (2008).

Least Angle Regression

Least angle regression (LAR) utilizes the piece-wise linear form of LASSO solution by working step by step to find the solution path. It uses similar strategy to the forward selection 2.1.2, where one variable is added at a time.

Starting with a null model, we set the centred response variable as the residual vector. Define $\mathcal{A}_k = \{j|\hat{\beta}_j(\lambda_k) \neq 0\}$ as the active set of variables at the beginning of the kth step and $\boldsymbol{\beta}_{\mathcal{A}_k}$ be the coefficient vector for the variables in the active set at step k. A variable is then added to the active set if it has the highest correlation with the response variable. However, rather than fitting the model with the selected variable, LAR moves the corresponding coefficient estimates of the selected variable in the direction of the least squares coefficient,

$$\delta_k = (\mathbf{X}_{\mathcal{A}_k}^T \mathbf{X}_{\mathcal{A}_k})^{-1} \mathbf{X}_{\mathcal{A}_k}^T \mathbf{r}_k, \tag{2.13}$$

given the current residual is

$$\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{\mathcal{A}_k} \boldsymbol{\beta}_{\mathcal{A}_k}.$$

Thus, the coefficient vector now becomes

$$\boldsymbol{\beta}_{\mathcal{A}_k}(\lambda) = \boldsymbol{\beta}_{\mathcal{A}_k} + \lambda \delta_k.$$

The form of direction (2.13) keeps the correlation tied and decreasing in absolute value. The estimates continue to move in the direction until another variable becomes equally correlated with the residual. The term is then added to the active set, and their correlations are tied and decreasing. The process continues until all the variables are in the model. The steps are summarized in Algorithm 2.2.1.

Algorithm 2.2.1 (Least Angle Regression (LAR))**.**

1. *Standardize predictors and start with residual* $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$ *and* $\boldsymbol{\beta} = \boldsymbol{\beta}$.
2. *Find the predictor* $\mathbf{X}_j$ *which most correlated with* $\mathbf{r}$.
3. *Move* β_j *away from 0 and towards the least-squares coefficient* $\langle \mathbf{X}_j, \mathbf{r} \rangle$, *until there exist another variable* $\mathbf{X}_k$ *which has similar correlation with the current residual as* $\mathbf{X}_j$.

4. *Move β_j and β_k in the direction defined by their joint least squares coefficient of the current correlation on $(\mathbf{X}_j, \mathbf{X}_k)$, until there exist another variable $\mathbf{X}_\ell$ has similar correlation with the current residual.*

5. *Continue the above process until all p predictors have been included to the model. After $min(N-1, p)$ steps, we obtain the full least squares solution.*

It is shown in Hastie (2008) that LAR produces a set of coefficient profiles which almost identical to the LASSO coefficient profiles. Thus, the LAR algorithm is modified to solve the LASSO problem and is called LAR-LASSO. The algorithm is given below.

Algorithm 2.2.2 (LAR-LASSO)**.**

1. *Apply LAR algorithm up to Step 4.*

2. *If a non-zero coefficient reduces to zero, remove the variable from the current active set and recompute the joint least squares direction.*

3. *Continue the above process until all p predictors have been considered.*

The LAR-LASSO algorithm is extremely efficient in solving LASSO problem. It only require the same order of computation as fitting a single least squares fit with p predictors (Hastie, 2008).

2.2.3 BRidge

Frank and Friedman (1993) discuss a generalization of ridge and LASSO regression which uses L_q-norm. This method was not given an explicit name in the original paper but was referred to as 'bridge' in (Tibshirani, 1996), and still uses the reference up until now.

For $q \geq 1$, BRidge regression solves the following problem,

$$\hat{\beta}^B = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}. \tag{2.14}$$

Equivalently,

$$\hat{\beta}^B = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 \right\} \qquad \text{subject to} \sum_{j=1}^{p} |\beta_j|^q \leq t. \tag{2.15}$$

It is immediate from (2.14) and (2.15) that $q = 1$ corresponds to LASSO while $q = 2$ gives the ridge regression.

In the case of an orthonormal design, the BRidge estimates can be derived in similar way as the ridge and LASSO to be

$$\hat{\beta}_j^B = \hat{\beta}_j^o - \lambda q |\hat{\beta}_j^B|^{q-1} \text{sign}(\hat{\beta}_j^B)/2 \tag{2.16}$$

Figure 2.2 shows the shrinkage effect of BRidge regression. As shown in 2.2(b), when $q \in (1, 2)$, BRidge regression penalize small least square estimates by a higher scale compared to the penalty imposed on large estimates. The opposite holds for $q > 2$ as indicated in Figure 2.2(a). In summary, BRidge regression with large values of q ($q \geq 2$) tends to retain small parameters while small

values of q ($q < 2$) tends to shrink small parameters to zero. It is important to note that for all $q \neq 1$, BRidge only shrinks the coefficients, hence unable to perform variable selection. Based on Fan and Li (2001), the only variation of BRidge regression which has both continuous solution and also a thresholding rule (i.e., able to set estimated coefficients to zero) in orthogonal case is the LASSO ($q = 1$).

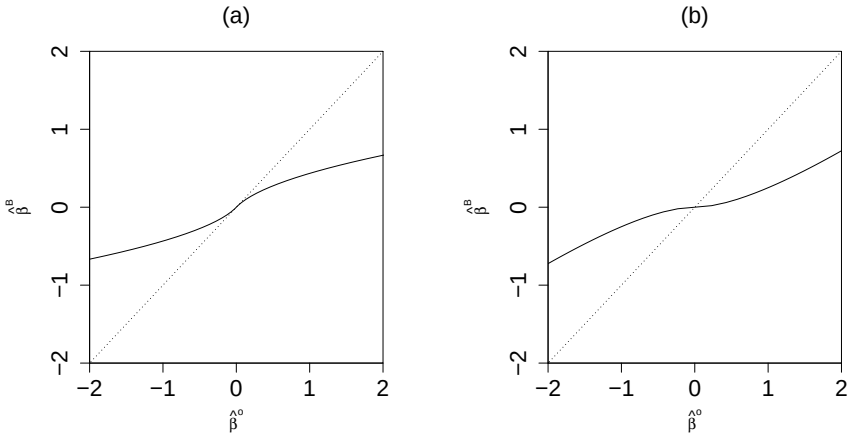


Figure 2.2: Plot of the solution for the BRidge regression with $\lambda = 3$ and (a) $q = 3$, and (b) $q = 1.5$. The 45° dotted lines represents the full least square estimates.

2.2.4 Non-negative Garrote

In this part, we will be discussing the method that has been a motivation for the construction of LASSO which is the non-negative garrote. The non-negative garrote was introduced by Breiman (1995) and the problem is formulated as a scaled version of the least squares estimate 1.1. Define $d(\lambda) = (d_1(\lambda), d_2(\lambda), ..., d_p(\lambda))'$ as the shrinking factor. Then, the optimization problem is given as

$$\hat{\beta}^{nn} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} Z_j d_j \right) + n\lambda \sum_{j=1}^{p} d_j \right\}, \qquad \text{subject to } d_j > 0, \ \forall j \tag{2.17}$$

where the scaled regression variable, $Z_j = X_j \beta_j^o$. Here $\lambda > 0$ is still the tuning parameter. Since the problem is the scaled version of least squares, the coefficient estimates for non-negative garrote can be expressed in terms of the least squares estimates namely,

$$\hat{\beta}_j^{nn} = d_j(\lambda)\hat{\beta}_j^o$$

Under the orthogonal design case, the minimizer for (2.17) has a closed form given by

$$d_j(\lambda) = \left(1 - \frac{\lambda}{\hat{\beta}_j^{o2}} \right)_+ \tag{2.18}$$

for $j = 1, 2, ..., p$. In terms of shrinkage effect, non-negative garrote applies smaller penalty on large estimates as the shrinkage factor will be close to 1 by definition of (2.18). For a redundant parameter, the least squares estimate

tends to be lower, hence the shrinkage factor is larger increasing the possibility of setting the corresponding estimate to zero.

As mentioned at the start of this chapter, non-negative garrote is one of the penalized method with concave penalty. It is the first penalty of this kind. Unfortunately, it suffers from a number of bad properties. Its explicit dependent on the least squares makes it less reliable when the sample size is small as the least squares perform poorly in that setting. It also has inconsistent variable selection. It is rapidly abandoned for a more consistent and reliable method discussed next.

2.2.5 SCAD

The problem with LASSO is that for large coefficients, the penalty is linear in size of the regression coefficient as seen in Figure 2.1. This results in biased estimates for the corresponding coefficients. To that end, Fan and Li (2001) proposed a non-convex penalty function referred to as the *"smoothly clipped absolute deviation"* (SCAD) penalty. The SCAD estimate solves the following problem,

$$\hat{\beta}^s = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 + p_S(\beta_j) \right\}, \tag{2.19}$$

where the penalty term, $p_S(\beta_j)$ is given by,

$$p_S(\beta_j) = \begin{cases} \lambda|\beta_j| & \text{if } |\beta_j| \leq \lambda, \\ \frac{2a\lambda|\beta_j| - \beta_j^2 - \lambda^2}{2(a-1)} & \text{if } \lambda \leq |\beta_j| \leq a\lambda, \\ \frac{\lambda^2(a+1)}{2} & \text{if } |\beta_j| \geq a\lambda. \end{cases} \tag{2.20}$$

For some $a > 2$ and $\lambda > 0$, the first derivative of the penalty function (2.20) is given as

$$p'_S(\beta_j) = \lambda \left\{ I(\beta_j \leq \lambda) + \frac{(a\lambda - \beta_j)_+}{(a-1)\lambda} I(\beta_j > \lambda) \right\}. \tag{2.21}$$

The SCAD penalty is continuously differentiable on $(-\infty, 0) \bigcup (0, \infty)$ but singular at 0 with zero derivative outside the range $[-a\lambda, a\lambda]$. This allows SCAD to set small coefficients to zero, while does not excessively penalized large coefficients. Thus, SCAD can produce sparse set of solution as well as unbiased estimations for large coefficients. It is shown in Fan and Li (2001) that SCAD has the oracle properties:

- Consistency in variable selection.
- Asymptotic normality.

In simple terms, oracle property means that the selected model performs as excellent as if the true underlying model would be known. This suggests that SCAD is a great variable selection method which potentially outperforms the previously discussed methods in most data settings. It is also mentioned in Fan and Li (2001) that this outstanding performance holds true only when the noise level, determined by σ, is low. Otherwise, other methods could potentially have better variable selection and prediction accuracy compared to SCAD.

Under the orthonormal design case, the SCAD solution is given as

$$\hat{\beta}_j^s = \begin{cases} \text{sign}(\hat{\beta}_j^o)(|\hat{\beta}_j^o| - \lambda)_+ & \text{if } |\hat{\beta}_j^o| < 2\lambda, \\ (a-1)\hat{\beta}_j^o - \text{sign}(\hat{\beta}_j^o)a\lambda/(a-2) & \text{if } 2\lambda \leq |\hat{\beta}_j^o| \leq a\lambda, \\ \hat{\beta}_j^o & \text{if } |\hat{\beta}_j^o| > a\lambda. \end{cases} \tag{2.22}$$

This solution is plotted in Figure 2.3(a). Based on the plot, we can see that for the first interval, $|\hat{\beta}_j^o| < 2\lambda$, heavier penalty were imposed on the coefficient estimates while estimates which are lower than $\lambda = 3$, are set to zero. Smaller penalty were applied to estimates in the second interval, $2\lambda \leq |\hat{\beta}_j^o| \leq a\lambda$, while coefficient estimates which are larger than $a\lambda = 11.1$ are not penalized, hence being equal to the least squares estimates. Thus, its geometry supports our discussion.

The thresholding rule in (2.22) involves two unknown parameters λ and a. Theoretically, the best pair (λ, a) could be obtained using two dimensional grids search using some criteria such as cross-validation or generalized cross-validation. Due to its non-convex penalty, implementing SCAD can be computationally expensive. Based on Bayesian statistical point of view and simulation studies, Fan and Li (2001) suggested $a \approx 3.7$ as a good choice when the dimension of predictor variables is less than 100. In fact, choosing $a = 3.7$ works similarly to value of a chosen by generalized cross-validation (GCV) method discussed in Chapter 3. Thus, we simply use this value in Chapter 4 and 5.

2.2.6 Adaptive LASSO

The adaptive LASSO or adaLASSO is an improved version of the original LASSO (Tibshirani, 1996). This method was proposed by Zou (2006) with the aim of identifying whether L_1-penalty can also attain the oracle properties mentioned in Fan and Li (2001).

The adaptive LASSO adds in *adaptive weights*, w_j which penalize coefficients estimates in the L_1-penalty differently. The estimates using this method are given by

$$\hat{\beta}^A = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{n} y_i - \sum_{j=1}^{p} X_{ij}\beta_j + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}. \tag{2.23}$$

Note that (2.23) is a convex optimization problem with an L_1 constraint. Thus, similar to LASSO, adaptive LASSO can also be solved using cyclic coordinate descent discussed previously. In a multiple linear regression setting, it is reasonable to define the weight vectors as $w_j = 1/|\hat{\beta}_j^o|^\gamma$, where $\hat{\beta}_j^o$ is the ordinary least squares estimate with $\gamma > 0$ (Zou, 2006). If $w_j = 1$, then adaLASSO is equals to LASSO regression.

Consider the orthonormal design case. Then, the adaptive LASSO estimates can easily be derived, in similar way to LASSO, to be

$$\hat{\beta}_j^A = \text{sign}(\hat{\beta}_j^o)\left(|\hat{\beta}_j^o| - \frac{\lambda}{|\hat{\beta}_j^o|^\gamma} \right)_+ . \tag{2.24}$$

From Figure 2.3 (b), we can see that the solution for adaptive LASSO behaves in a similar way to the SCAD solution. Small coefficients are set to zero while the remaining large coefficients are not excessively penalized.

Thus, adaptive LASSO is able to yield consistent parameter estimates while retaining the convexity property of LASSO (Hastie, 2008). On top of that, Zou (2006) also proves that besides SCAD, adaptive LASSO is also an oracle procedure.

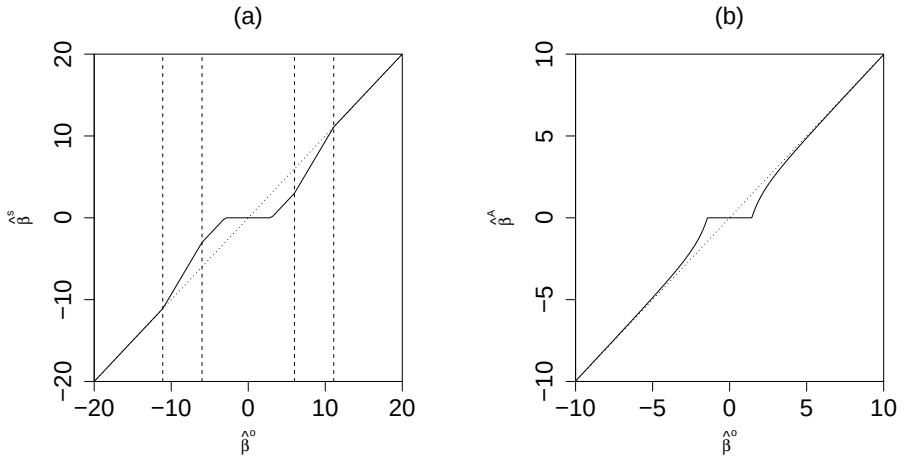


Figure 2.3: Plot of the thresholding functions with $\lambda = 3$ and (a) $a = 3.7$ for SCAD, and (b) $\gamma = 2$ for adaptive LASSO. The vertical line in (a) represents the value of $\pm 2\lambda = \pm 6$, and $\pm a\lambda = \pm 11.1$. The 45° dotted lines represents the full least square estimates.

2.2.7 Elastic net

Zou and Hastie (2005) pointed out some limitations of LASSO which are,

(a) In the high-dimensional setting ($p \gg n$), LASSO selects at most n variables due to the nature of convex optimization problem.

(b) When a group of variables has high pairwise correlations, LASSO tends to select only one variable from the groups and ignore the rest.

(c) In the usual $n \gg p$ setting, if there are high correlations between predictors, ridge has been empirically observed to perform better than LASSO (Tibshirani, 1996).

For those reasons, Zou and Hastie (2005) established a new method called the elastic net. Elastic net is an ideal variable selection method in scenarios (a) and (b), especially in genomics and other areas of computational biology which involves microarray data. It should also deliver better prediction performance than LASSO in scenario (c).

The derivation of elastic net starts with the naive elastic net. Consider a data set with n observations with p predictors. Let $\mathbf{y} = (y_1, ..., y_n)^T$ be the response and $\mathbf{X}$ be the design matrix. After a location and scale transformation, we can assume that the response are centred and the predictors are standardized. For any fixed non-negative λ_1 and λ_2, the naive elastic net solves the following

optimization problem.

$$\hat{\beta}^{NEN} = \underset{\beta}{\operatorname{argmin}}\left\{ \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j X_{ij})^2 + \lambda_2 \sum_{j=1}^{p}|\beta_j|^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j|_1\} \right\}. \quad (2.25)$$

Define $\alpha = \lambda_2/(\lambda_1 + \lambda_2)$, then (2.25) can be written as

$$\begin{aligned} \hat{\beta}^{NEN} = \underset{\beta}{\operatorname{argmin}}&\left\{ \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j X_{ij})^2 \right\} \\ &\text{subject to} \quad (1-\alpha)\sum_{j=1}^{p}|\beta_j| + \alpha \sum_{j=1}^{p}|\beta_j|^2 \leq t, \end{aligned} \quad (2.26)$$

where the function $(1-\alpha)\sum_{j=1}^{p}|\beta_j| + \alpha\sum_j |\beta_j|^2$ is the elastic net penalty. Essentially, this is a convex combination of the LASSO and ridge tuning parameters, λ_1 and λ_2 respectively. Similar to ridge, the penalty function if strictly convex for all $\alpha > 0$. Also, the singularity at 0 allows naive elastic net to produce sparse solutions like the LASSO. Zou and Hastie (2005) highlight the strict convexity of the constraint allows the method to group highly correlated predictors given their effects are equal in size. The LASSO penalty is convex but not strictly convex. Hence, it does not have the grouping effect.

When $\alpha = 1$, naive elastic net becomes the ridge regression while $\alpha = 0$ corresponds to LASSO. Specifically, the elastic net with $\alpha = \epsilon$ for some small $\epsilon > 0$ performs much like the LASSO, but removes any degeneracies and erratic behaviours caused by extreme correlations. As α decreases from 1 to 0, the sparsity –the number of coefficients equal to zero– of solution to (2.25) increases monotonically from 0 to the sparsity of the LASSO solution.

As shown by Zou and Hastie (2005), minimizing (2.25) is equivalent to solving an equivalent LASSO problem on augmented data. This allows the naive elastic net to share the computational advantage of LASSO. Naive elastic net is also shown to overcome limitations of LASSO in problem (a) and (b) mentioned above.

Unfortunately, by imposing double amount of shrinkage, naive elastic net adds in unnecessary extra bias. This makes it performs less satisfactorily compared to LASSO or ridge regression alone. As a result, the corrected version of elastic net was introduced to undo the extra shrinkage. The elastic net estimates are defined as a re-scaled naive elastic net coefficients.

$$\hat{\boldsymbol{\beta}}^{EN} = (1+\lambda_2)\hat{\boldsymbol{\beta}}^{NEN}.$$

Given data $(\mathbf{y}, \mathbf{X})$ and (λ_1, λ_2), the elastic net estimates are given by

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{EN} &= \underset{\beta}{\operatorname{argmin}}\Big\{(1+\lambda_2)\big[\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\beta\|_2^2\big]\Big\} \\ &= \underset{\beta}{\operatorname{argmin}}\left\{\boldsymbol{\beta}^T\Big(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2}\Big)\boldsymbol{\beta} - 2\boldsymbol{y}^T\mathbf{X}\boldsymbol{\beta} + \lambda_1|\boldsymbol{\beta}|_1\right\}. \end{aligned}$$

This allows elastic net to preserves the variable selection properties of the naive elastic net but also undo shrinkage.

2.2.8 Group LASSO

Group LASSO was introduced by Yuan and Lin (2006) as an extension of LASSO, with similar aim to elastic net, which is to have the *grouping effect*.

Suppose that p predictors are divided into L groups, with p_ℓ be the number of predictors in group $\ell \in \{1, ..., L\}$. We denote $\mathbf{X}_\ell \in \mathbb{R}^{n \times p_\ell}$ to represent the predictors corresponding to the ℓ-th group, with corresponding coefficient vector $\beta_\ell \in \mathbb{R}^{p_\ell}$. Similar to settings in previous methods, we assume that $\mathbf{y}$ and $\mathbf{X}$ have been centred. Then, the group LASSO estimate solves the convex optimization problem

$$\hat{\beta}^G = \underset{\beta}{\operatorname{argmin}} \Big\{ \|\mathbf{y} - \sum_{\ell=1}^{L} \mathbf{X}_\ell \beta_\ell\|_2^2 + \lambda \sum_{\ell=1}^{L} \sqrt{p_\ell} \|\beta_\ell\|_{K_\ell} \Big\}, \tag{2.27}$$

where the p_ℓ accounts for the varying group sizes, and $\|\beta_\ell\|_{K_\ell} = (\beta_\ell^T K_\ell \beta_\ell)^{1/2}$ is the K_ℓ- quadratic norm and K_ℓ is a symmetric positive definite matrix. When $p_\ell = 1$, the group LASSO simply becomes the original LASSO.

Based on Yuan and Lin (2006), a necessary and sufficient condition for a vector $\beta = (\beta_1^T, \beta_2^T, ..., \beta_L^T)^T$ to be a solution for (2.27) is

$$\mathbf{X}_\ell^T(\mathbf{y} - \mathbf{X}\beta) + \frac{\lambda \beta_\ell \sqrt{p_\ell}}{\|\beta_\ell\|} = \mathbf{0}, \qquad \forall \beta_\ell \neq \mathbf{0}, \tag{2.28}$$

$$\| - \mathbf{X}_\ell^T(\mathbf{y} - \mathbf{X}\beta)\| \leq \lambda \sqrt{p_\ell}, \qquad \forall \beta_\ell = \mathbf{0}. \tag{2.29}$$

We follow the method used in Yuan and Lin (2006), where we find the solution to (2.28) and (2.29) based on the idea of *shooting algorithm* (refer (Fu, 1998) for full details). We first consider the case $\beta_\ell \neq 0$. Expanding equation (2.28) gives us

$$-\mathbf{X}_\ell^T \mathbf{y} + \mathbf{X}_\ell^T \mathbf{X}_\ell \beta_\ell + \sum_{i \neq \ell} \mathbf{X}_\ell^T \mathbf{X}_i \beta_i + \frac{\lambda \beta_\ell \sqrt{p_\ell}}{\|\beta_\ell\|} = \mathbf{0}.$$

Grouping terms with β_ℓ

$$\Big(\mathbf{X}_\ell^T \mathbf{X}_\ell + \frac{\lambda \sqrt{p_\ell}}{\|\beta_\ell\|}\Big)\beta_\ell = \mathbf{X}_\ell^T \Big(\mathbf{y} - \sum_{i \neq \ell} \mathbf{X}_i^T \beta_i\Big) =: S_\ell,$$

where $S_\ell = \mathbf{X}_\ell^T(\mathbf{y} - \mathbf{X}\beta_{-\ell})$, with $\beta_{-\ell} = (\beta_1^T, ..., \beta_{\ell-1}^T, \mathbf{0}^T, \beta_{\ell+1}^T, ..., \beta_L^T)$. So,

$$\beta_\ell = \Big(\mathbf{X}_\ell^T \mathbf{X}_\ell + \frac{\lambda \sqrt{p_\ell}}{\|\beta_\ell\|}\Big)^{-1} S_\ell. \tag{2.30}$$

In the orthogonal design case, it can easily be shown that

$$\|\beta_\ell\| = \Big(1 + \frac{\lambda \sqrt{p_\ell}}{\|\beta_\ell\|}\Big)^{-1} \|S_\ell\| = \|S_\ell\| - \lambda \sqrt{p_\ell}.$$

This allows us to rewrite (2.30) as

$$\beta_\ell = \Big(1 + \frac{\lambda \sqrt{p_\ell}}{\|S_\ell\| - \lambda \sqrt{p_\ell}}\Big)^{-1} S_\ell = \Big(1 - \frac{\lambda \sqrt{p_\ell}}{\|S_\ell\|}\Big) S_\ell. \tag{2.31}$$

Now consider the case where $\beta_\ell = 0$. Using (2.29) and the definition of S_ℓ, we have the condition simplifies to

$$\| - \mathbf{X}_\ell^T(\mathbf{y} - \mathbf{X}\beta)\| = \|S_\ell\| \leq \lambda\sqrt{p_\ell} \qquad \forall \beta_\ell = \mathbf{0}.$$

Thus, in order to have non- zero solution, we require $\|S_\ell\| > \lambda\sqrt{p_\ell}$. Combining these two conditions, we have the solution to group LASSO problem which satisfies (2.28) and (2.29) as

$$\hat{\beta}_\ell = \Big(1 - \frac{\lambda\sqrt{p_\ell}}{\|S_\ell\|}\Big)_+ S_\ell. \tag{2.32}$$

The solution to (2.27) can then be obtained by iteratively applying equation (2.32) for $\ell = 1, ..., L$.

As the name suggests, group LASSO acts like the LASSO at group level. It either retains a group of predictors or remove all members of the group. Hence, group LASSO promotes sparsity between the mutually exclusive groups but not within groups (Friedman et al., 2010a).

The method of solving (2.27) discussed above is very stable and usually reaches reasonable convergence tolerance within a few iterations. Unfortunately, the method suffers high computational costs for large number of predictors. Friedman et al. (2010a) pointed out the limitation of the algorithm in which it assumes model matrices in each group are orthonormal. This is however, not always the case.

Due to all the limitations of group LASSO, Friedman et al. (2010a) proposed an improved method called the *sparse group LASSO* which able to yield sparsity at both group and individual levels and also a more useful algorithm which works in non-orthonormal model matrices. The sparse group LASSO criterion is given by

$$\hat{\beta}^{SG} = \underset{\beta \in \mathrm{R}^p}{\operatorname{argmin}} \Big\{ \|\mathbf{y} - \sum_{\ell=1}^{L} \mathbf{X}_\ell \beta_\ell\|_2^2 + \lambda_1 \sum_{\ell=1}^{L} \sqrt{p_\ell} \|\beta_\ell\|_2 + \lambda_2 \|\beta\|_1 \Big\}, \tag{2.33}$$

where $\beta = (\beta_1, \beta_2, ..., \beta_\ell)$ is the entire parameter vector. Note that the expression above is a sum of convex functions and hence convex itself. When $\lambda_2 = 0$, we have the group LASSO and $\lambda_1 = 0$ corresponds to LASSO.

Chapter 3

Model assessment and selection

As we have discussed in Chapter 2, we want the variable selection methods to perform well in predicting the response for independent test data. The process of evaluating a model's performance is called model assessment and is extremely crucial as it gives a measure of quality of the chosen model. We have seen that subset selection methods result in a set of models with varying size. On the other hand, shrinkage methods produces a set of models corresponding to each value of the tuning parameter, λ. Thus, a best-fitting model need to be selected from these set of models with varying complexity. This problem is referred to as model selection (James et al., 2014).

The most common measure to tackle both of these problems is the prediction error. Section 3.1 outlines the relation of prediction error with bias, variance and model complexity. However, when dealing with real-life data, the true model is unknown. Hence, computing the prediction error directly is impossible. One approach is to use an independent validation data set. Unfortunately, this method is only feasible if we have enough data to set aside as an independent validation data set. In more realistic case, where extra data are unavailable, we can estimate the prediction error from the training set. Although training error often will be lower than the true prediction error, and is referred to as the *optimism* of training error. We will not be going into the details of this problem, but explores the method to offset this problem using the information criteria discussed in Section 3.2, which helps to provide better estimate of the prediction error. Apart from altering the error form using information criteria, we can also achieve better prediction error estimates via resampling methods. In Section 3.3, we discussed some variation of cross-validation (CV).

3.1 Prediction error

Prediction error is defined as the average error in prediction of $\mathbf{y}$ given $\mathbf{X}$ from the test data set. In other words, it gives a measure of how well our model performs on unseen data so that we can assess the quality of the model. It also helps in model selection as it acts as a mean for comparing different models.

Consider the case X random. Then our training set, $\mathcal{T} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), ..., (\mathbf{X}_k, y_k)\}$, for some $k < n$, is assumed to be randomly sampled from the parent distribution $(\mathbf{X}, \mathbf{y})$. Suppose $\hat{\mu}(\mathbf{X})$ is the predictive model obtained from the training set. Then, the loss function for measuring the discrepancies between $\mathbf{y}$ and $\hat{\mu}(\mathbf{X})$, denoted as $L\big(\mathbf{y}, \hat{\mu}(\mathbf{X})\big)$, in terms of the squared error is given as

$$L\big(\mathbf{y}, \hat{\mu}(\mathbf{X})\big) = \sum_{i=1}^{k} \big(y_i - \hat{\mu}(X_i)\big). \tag{3.1}$$

The training error is simply the average loss over the training sample, $\mathcal{T}$,

$$\mathrm{TE}\Big(\hat{\mu}(\mathbf{X})\Big) = \frac{1}{k} L\big(\mathbf{y}, \hat{\mu}(\mathbf{X})\big) = \frac{1}{k} \sum_{i=1}^{k} \big(y_i - \hat{\mu}(X_i)\big) \qquad \text{for } (\mathbf{y}, \mathbf{X}) \in \mathcal{T}. \tag{3.2}$$

Let us denote an independent test set as $\mathcal{D} = \{(\mathbf{X}_{k+1}, y_{k+1}), ..., (\mathbf{X}_n, y_n)\}$. Then, the test error for a specific training set is the prediction error over the set $\mathcal{D}$ defined as

$$\mathrm{PE}_{\mathcal{T}}\Big(\hat{\mu}(\mathbf{X})\Big) = E\Big[L\big(\mathbf{y}, \hat{\mu}(\mathbf{X})\big)|\mathcal{T}\Big] = E\Big[(\mathbf{y} - \hat{\mu}(\mathbf{X})^2)|\mathcal{T}\Big] \qquad \text{for } (\mathbf{y}, \mathbf{X}) \in \mathcal{D}. \tag{3.3}$$

The quantity that we are interested in is the expected prediction error, defined as

$$\mathrm{PE}\Big(\hat{\mu}(\mathbf{X})\Big) = E\Big[\mathrm{PE}_{\mathcal{T}}\Big(\hat{\mu}(\mathbf{X})\Big)\Big] \tag{3.4}$$

The expected prediction error of a regression fit $\hat{\mu}(\mathbf{X})$ can be decomposed into three components,

$$\begin{aligned}
\mathrm{PE}\Big(\hat{\mu}(\mathbf{X})\Big) &= E\Big[\big(\mathbf{y} - \hat{\mu}(\mathbf{X})\big)^2\Big] \\
&= E\Big[\mathbf{y} - E(\mathbf{y}|\mathbf{X})\Big]^2 + E\Big[\hat{\mu}(\mathbf{X}) - \mu(\mathbf{X})\Big]^2 \\
&= \mathrm{Var}(\mathbf{y}) + E\Big[\hat{\mu}(\mathbf{X}) - E\big(\hat{\mu}(\mathbf{X})\big)\Big]^2 + \Big[E\big(\hat{\mu}(\mathbf{X})\big) - \mu(\mathbf{X})\Big]^2 \\
&= \sigma^2 + \mathrm{Var}\big(\hat{\mu}(\mathbf{X})\big) + \mathrm{Bias}^2\big(\hat{\mu}(\mathbf{X})\big).
\end{aligned}$$

The formulation above is referred to as the *bias-variance decomposition*. The first term is the unavoidable variance of new response around the true mean $\mu(\mathbf{X})$. The second term is the variance of $\hat{\mu}(\mathbf{X})$ around its mean, and the last term is the squared bias which is the average of difference between our estimates and its true mean.

The sum of the last two terms in the bias-variance decomposition, namely the squared bias and the variance, is referred to as the *mean-squared error*(MSE) of an estimated model $\hat{\mu}(\mathbf{X})$, formally written as

$$\mathrm{MSE}(\hat{\mu}(\mathbf{X})) = \mathrm{Var}\big(\hat{\mu}(\mathbf{X})\big) + \mathrm{Bias}^2\big(\hat{\mu}(\mathbf{X})\big). \tag{3.5}$$

Since a large portion of the expected prediction error is the MSE, simulation results and data analysis often presented in terms of the MSE rather than the expected prediction error.

In general, a more complex model $\hat{\mu}(\mathbf{X})$, tends to fit noise in the training data. This problem is called *overfitting*. This reduces the squared bias but consequently increases the variance. Hence, the model performs poorly when predicting responses from a new set of data i.e., the test set. On the contrary, a too simple model tends to underfit a data. The variance will be low but the model will be very biased. Underfitted model will also not generalize well to new data. Our aim is to find the proper model complexity which gives a balanced trade-off between bias and variance resulting to minimum expected prediction error.

As mentioned earlier in this chapter, prediction error can only be computed given the true model is known. This, is usually not the case. A training error with expected optimism can be used as an estimate of the prediction error (Hastie, 2008). This quantity represents the estimate of in-sample error, and is expressed in terms of the training error 3.2 as

$$\widehat{\mathrm{Err}}_{in} = \mathrm{TE}\big(\hat{\mu}(\mathbf{X})\big) + \hat{\omega}, \tag{3.6}$$

where $\hat{\omega}$ is the estimate of the average optimism. Hence, we next discuss some of the available information criteria which equivalent to the expected of the in-sample error estimates.

3.2 Information criteria

Information criteria is used as an approximation to validation step when there is insufficient data to split into three parts; training, validation and test set. Besides their use in model selection, it also helps in assessing the final chosen model.

The general setup to select the best model of size $k \leq p$ starts with calculating the selected criterion for models of size $k = 1, 2, ..., p$. Then, the model corresponding to the *'best'* value is selected. Notice that we shift from using the RSS and R^2 in Step 2 to using information criteria in Step 3 of Algorithm 2.1.1 for subset selection methods. This is because the RSS is not a suitable measure to compare models with different sizes since it always improve (decreases for RSS, and increase for R^2) as more variables are added into the model. Hence, we use the information criteria which penalizes this action of adding covariates which does not adequately improve the fit.

Given the error follows Normal distribution and models are nested, F-statistics or the likelihood ratio test can be used to compared the models. These methods will not be discussed in this report. On the other hand, information criteria are more versatile in the sense that they are applicable for both nested and non-nested models.

3.2.1 Adjusted R^2

Recall from Chapter 1, R^2 is defined as 1-RSS/TSS. Due to the problems with RSS and R^2 mentioned above, we can use the adjusted R^2 as an alternative. The adjusted R^2 is defined as

$$R^2_{\text{adj}} = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}.$$

The adjusted R^2 takes into account the model complexity by adjusting the RSS and TSS based on their respective degree of freedom. Similar to the usual R^2, a large value of adjusted R^2 indicates a model has a small test error (James et al., 2014). The measure tells us the proportion of variation explained by independent variables that actually affect the dependent variable. Hence, it penalizes the regression model if we add in terms that do not improve the fit of model. Thus, in theory, the model with the largest adjusted R^2 will have only the correct variables with no noise in the model.

3.2.2 Mallow's C_p

Mallow's C_p is defined for a model containing k predictors as

$$C_p = \frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2), \tag{3.7}$$

where $\hat{\sigma}^2$ is the estimated error variance, obtained from a low-bias model (Hastie, 2008). A small value of C_p corresponds to a small mean squared errors for the fitted values, whilst a value of C_p close to $C_p = k = p$ corresponds to a model with small bias. The choice between the two options, low MSE or small bias depends on the objective of study and user's preference.

It is important to note that the second term $2k\hat{\sigma}^2$ in (3.7) can be viewed as the penalty on the RSS which aims to solve the fact that RSS tends to underestimate the prediction error. As RSS will continue to decrease when more variables are added, the penalty, on the contrary, will increase. This provides a more reliable way to measure the model performance as the addition of a new variable need to provide a sufficient decrease in the RSS to offset the penalty.

3.2.3 Akaike's Information Criterion (AIC)

If we assume the errors follows Normal distribution, then we can use measures based on the likelihood function. Given the assumption of normality, we have

$$y \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}).$$

The likelihood therefore comes directly from the definition of multivariate normal density

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2; \mathbf{y}) &= f(\mathbf{y}|\boldsymbol{\beta}, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{ \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}. \end{aligned}$$

Taking the log of the likelihood above gives,

$$\ell(\boldsymbol{\beta}) = \ln\big(L(\boldsymbol{\beta}, \sigma^2; \mathbf{y})\big) = -\frac{n}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Maximising the log-likelihood with respect to $\boldsymbol{\beta}$ is equivalent to minimizing the expression $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. Thus, under the assumption errors are Gaussian, the maximum likelihood coincides with least squares estimates. In this case, we can write the AIC as

$$\text{AIC} = -2\ell(\hat{\boldsymbol{\beta}}) + 2k = \frac{\text{RSS}}{\sigma^2} + 2k \tag{3.8}$$

where, for simplicity, we omitted the constant $n\ln(2\pi\sigma^2)$ which does not depend on the model parameters. This form of AIC is proportional to the Mallow's C_p (James et al., 2014). The first term of AIC is called the deviance which is a goodness-of-fit measure while the second term in (3.8) corresponds to the penalty term for AIC. Note that the penalty term does not depend on the sample size n. As a result, smaller amount of penalty is applied to the deviance, allowing more variables to enter the model. Thus, AIC tends to select model with the best prediction accuracy but does not guarantee the true covariates are included in the model. To use the AIC for model selection, we simply select the model with the lowest value of AIC.

3.2.4 Bayesian Information Criterion (BIC)

BIC is similar to AIC in the sense that we estimate the expected in-sample error (3.6) by penalizing the deviance obtained from maximum likelihood function. For least squares model with k predictors, the BIC is defined as

$$BIC = -2\ell(\hat{\boldsymbol{\beta}}) + k\,\ln(n) = \frac{\text{RSS}}{\sigma^2} + k\,\ln(n) \tag{3.9}$$

The BIC is proportional to AIC, with the factor 2 in AIC replaced by $\ln(n)$ in BIC. Considering $\ln(n) > 2$ for $n > e^2 = 7.389$, which holds most of the time, the BIC impose much heavier penalty on complex model, encouraging simpler models in the selection process. Thus, if our focus is on model selection, we prefer model with lower BIC.

3.3 Resampling methods

Resampling methods involve repeatedly drawing random samples from the training set and refitting each samples from a model to obtain a desired information. These methods can be used to measure accuracy, such as calculating the standard errors of estimates, and also performs model selection and assessment. As it involves iterating the process of fitting a model, resampling methods can be computationally expensive especially if the model is complex to begin with. Two of the most commonly used resampling methods are cross-validation and bootstrapping. In this section, we will only be focusing on some variation of cross-validation which will be used in Chapter 4 and 5.

3.3.1 Cross- Validation

As mentioned earlier in this chapter, it is unlikely to have a designated test set to estimate the test error rate or an independent validation set for the estimation of tuning parameter. Due to its simplicity and reasonable accuracy, cross-validation is a popular method to compute those estimates. The basic intuition behind cross-validation is to repeatedly perform the process of splitting the training data into several parts, holding out a single subset from the fitting process. We then apply any desired statistical measure on the held out subset.

K-fold Cross- Validation

K-fold CV repeatedly split the data into K roughly-equal parts, where one part is used for prediction while the remaining parts are used to fit the model. For each $k \in \{1, 2, ..., K\}$, the k-th part is used for validation while the other $k-1$ are used for training. The algorithm can be formalized as follows,

Algorithm 3.3.1 (K-fold CV)**.**

1. *Randomly split the data into K parts.*
2. *For $k = 1, 2, ..., K$,*
   (a) *Fit the model using the training data set with the k-th data set removed and denote the set by $\hat{\mu}^{-k}(X)$.*
   (b) *Calculate the estimated prediction error for the k-th fold*

$$\widehat{\mathrm{PE}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (y_i - \hat{\mu}^{-k}(X_i)),$$

   where n_k denotes the number of data in the k-th split.
3. *Compute the CV estimate of the expected PE which is the average of $\widehat{\mathrm{PE}}_k$ over all values of k*

$$CV_K(\hat{f}) = \frac{1}{K} \sum_{i=1}^{K} \widehat{\mathrm{PE}}_k.$$

Typical choices are $K = 5$ or $K = 10$, as these values have been shown empirically to yield prediction error estimates that are not excessively biased or excessively variable (Kohavi, 1995). A unique case where $K = n$ is called the *leave-one-out* cross validation (LOOCV). In this case, model is fit using all but the i-th data during the i-th iteration. However, this a less popular choice due to its tendency to be computationally expensive when n is large.

K-fold CV can also be used to find the optimum value of tuning parameter for shrinkage methods. Supposed we have a set of models $\mu(x, \alpha)$ indexed be a tuning parameter α. Denote the model fit with the k-th part removed by $\hat{\mu}^{-k}(x, \alpha)$. Then CV criterion is defined as

$$\mathrm{CV}(\hat{\mu}, \alpha) = \frac{1}{n_k} \sum_{i=1}^{n_k} (y_i - \hat{\mu}^{-k}(X_i, \alpha)). \tag{3.10}$$

The criterion (3.10) above provides an estimate of the test error. Thus, our goal is to find $\hat{\alpha}$ which minimizes (3.10). The final model is then given in terms of the chosen value $\hat{\alpha}$, $\mu(X, \hat{\alpha})$.

Generalized cross-validation

Generalized cross-validation (GCV) is an approximation to LOOCV, for a linear fitting method under the squared-error loss. Linear fitting method refers to methods which we can write the estimated model as

$$\hat{\mathbf{y}} = \hat{\mu}(\mathbf{X}) = \mathbf{S}\mathbf{y}.$$

Here, $\mathbf{S}$ is an $n \times n$ matrix which depends on the input X_i but not on the response y_i. Under this setting, the GCV is given by

$$\text{GCV}(\hat{\mu}) = \frac{1}{n} \sum_{i=1}^{n} \frac{RSS}{(1 - \text{tr}(\mathbf{S})/n)^2} \tag{3.11}$$

where the trace of matrix $\mathbf{S}$, $\text{tr}(\mathbf{S})$ is the effective number of parameters.

Similarly, we find $\hat{\alpha}$ which minimizes $\text{GCV}(\hat{\mu}, \alpha)$ with similar adjustments discussed for K-fold CV. Some application of this method to optimize tuning parameters include Tibshirani (1996) and Fan and Li (2001).

Chapter 4

Simulation Study

In the following examples, we selected some of the methods discussed in Chapter 2 and compare the performance of each method with respect to the oracle procedure in different data settings. The selected methods are ordinary least squares, stepwise regression, ridge regression, the LASSO, elastic net, SCAD and group LASSO. The method used as baseline is the oracle least squares, that is, least squares using only the true non-zero parameters. Section 4.1 explains the measures we use to assess the performance of each method. A study of different examples of data settings are presented in Section 4.2.

For comparison purposes, we fit LASSO using both cyclical coordinate descent algorithm used in the R package `glmnet` and LAR-LASSO algorithm used in the R package `lars`. All the simulations are conducted using R codes which are available in Appendix A.

4.1 Performance measures

In the following simulation studies, each shrinkage method fits a regression model over a grid of values of regularization parameter λ. As discussed in Chapter 2, this produces a set of coefficients, $\hat{\boldsymbol{\beta}}(\lambda)$ with the corresponding model $\hat{\mu}_\lambda(\mathbf{X}) = \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)$ are fitted based on the training sample. Since we are simulating the data, the volume of data is not restricted. Hence, we choose to set aside an independent validation set, selected to be the same size as the training sample, to be used for model selection. We consider N data sets. Performance measures are computed for each data set, producing a set of N measures. Sample statistics is then applied to the set of output, giving a single value indicating the performance of each method. The following are the measures used to compare the performance of different methods.

4.1.1 Prediction accuracy

As mentioned in Chapter 3, a large portion of prediction accuracy is the mean-squared error. Thus, in the following simulation studies, the prediction accuracy is assessed by calculating the mean squared error of the predictions for each method. Since the true parameter vector and covariance matrix of the predictor variables are known, we can calculate the true MSE of prediction directly,

$$MSE = E[\hat{\mu}(\mathbf{X}) - \mu(\mathbf{X})]^2. \tag{4.1}$$

The accuracy is measured by taking the sample median over the N computed MSE from each data set.

4.1.2 Variable selection

Variable selection is assessed by looking at the estimated parameters which are included in the best, selected models. We use the following measures to assess selection performance.

1. the number of coefficient estimates correctly and incorrectly set to zero.
2. the number of coefficient estimates correctly and incorrectly not set to zero.
3. the median number of coefficient estimates not set to zero from all the datasets.

The number of coefficient estimates incorrectly not set to zero corresponds to the Type I error of the hypothesis $H_0 : \beta = 0$ against the alternative hypothesis $H_1 : \beta \neq 0$. The number estimates incorrectly set to zero corresponds to a type II error. We exclude the number of coefficients correctly not set to zero in the result presentation since in all cases, we found that all the methods are less likely to set coefficient estimates to zero. Thus, the number of coefficients correctly not set to zero is always equal to the true number of non-zero parameters.

4.2 Examples

In all examples, we considered various linear models of the form,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\varepsilon, \qquad \varepsilon \sim N(0, 1), \tag{4.2}$$

and grouped model,

$$\mathbf{y} = \sum_{\ell=1}^{L} \mathbf{X}_\ell \boldsymbol{\beta}_\ell + \sigma\varepsilon, \qquad \varepsilon \sim N(0, 1), \tag{4.3}$$

where the predictors are divided into L non-overlapping groups. Five examples are presented here. The first two explores the effect of varying the noise level, number of samples, and also correlation. The remaining examples create a grouped variable situations through (i) polynomial regression, (ii) specifying the correlation between predictors and (iii) using grouped model (4.3).

We follow similar methodology used in Zou and Hastie (2005). Within each example, our simulated data are split into training set, an independent validation set and an independent test set. The above models are fitted on training data only, and the validation set are used to select tuning parameters. We compute the test error (the mean-squared error) on the test data set. We use the notation $\cdot/\cdot/\cdot$ to describe the number of observations in the training, validation and test set respectively.

Example 1

In example 1, we consider moderate number of parameters with large effects. We simulated 50 data sets consisting n observations from the linear model (4.2) where $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$. The correct model used to fit the oracle is given by

$$f(X) = X_1\beta_1 + X_2\beta_2 + X_5\beta_5.$$

We let the number of observations for training and validation set, n be 20 and 60, so that we have 20/20/200 and 60/60/200 ratio of observations for training, validation and testing respectively. The predictor variables are related by an autoregressive, AR(1) correlation structure, $\text{corr}(X_i, X_j) = \rho^{|i-j|}$ as shown in Figure 4.1. Since the predictors have unit variance, the design matrix $\mathbf{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ with $\Sigma_{ij} = \rho^{|i-j|}$. The effect of noise is varied by using $\sigma \in \{1, 3, 6\}$ with corresponding signal-to-noise ratio SNR shown in Table 4.1

This example was studied in the original LASSO paper by Tibshirani (1996) and appears in a number of studies, including Fan and Li (2001), Zou (2006), and Zou and Hastie (2005) under various scenarios.

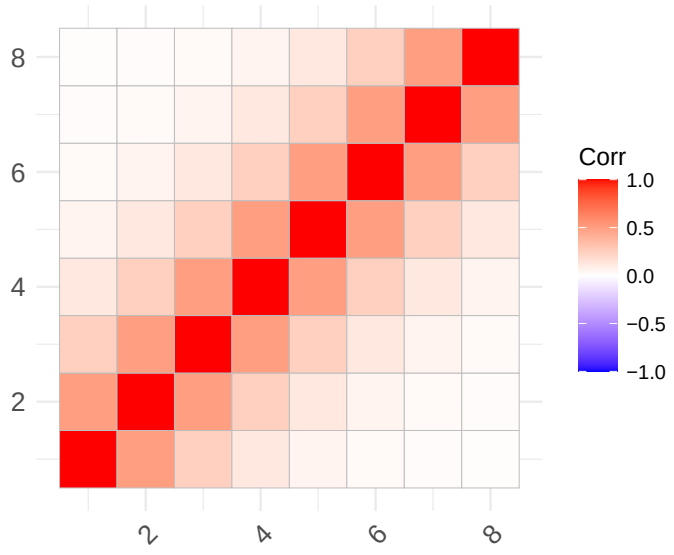


Figure 4.1: Plot of the AR(1) correlation matrix.

n	SNR		
	$\sigma = 1$	$\sigma = 3$	$\sigma = 6$
20	20.43	2.27	0.55
60	36.65	4.07	1.02

Table 4.1: SNR for the simulated data averaged over 50 data sets.

Table 4.2 shows the median MSE, the number of zero coefficients correctly and incorrectly set to zero, average number of incorrect non-zero coefficients and also the median non-zero coefficients for the aforementioned variable selection methods. For both sample sizes, we can see that when the noise level is high, elastic net consistently performs the best, followed by SCAD and group LASSO. The performance of LASSO using LAR algorithm is similar to the coordinate descent in which both performs moderately. Ordinary least squares, ridge and stepwise regression performs poorly. However, when the noise level is reduced, SCAD outperforms the elastic net and other penalizes least squares. It can also be seen from Table 4.2 that the performance of SCAD is as expected, to be as good as the oracle estimator as the sample size n increases.

We will exclude ridge and ordinary least squares from the following discussion as they did not performs variable selection. From Table 4.2, we can see that all the methods produces sparse solutions. The SCAD and stepwise regression tends to select the most accurate number of parameters while other methods tends to select more variables. Since our predictors are correlated, elastic net and group LASSO selects the most variables due to the grouping effect. Overall, SCAD performs the best in terms of prediction accuracy and also variable selection.

Example 2

In this example, we investigate the effect of correlation on the performance of variable selection methods when we have large number but small size of predictors. Given that we have more predictors, we choose to simulate 100 datasets with 60/60/200 observations. We define the beta coefficients as

$$\boldsymbol{\beta} = (\underbrace{0.2, ..., 0.2}_{20})^T$$

and the predictor variables are correlated by compound symmetry correlation structure,

$$\text{corr}(X_i, X_j) = \begin{cases} 1 & i = j, \\ \rho & i \neq j, \end{cases}$$

where the strength of correlation is varied by considering $\rho \in \{0.2, 0.9\}$. Figure 4.2 shows the plot of the correlation matrix. The effect of noise is fixed by setting $\sigma = 3$ with corresponding signal-to-noise ratio 1.692915 (for $\rho = 0.2$) and 3.28741 (for $\rho = 0.9$). Note that for this example, the oracle is equivalent to the ordinary least squares since the true number of non-zero parameters is the same as the total number of parameters. We also omitted the column for average number of incorrectly set as non-zero coefficients since in all cases, this is equals to zero.

Method	*n*	σ	*MSE*	*Avg. No. of 0 Coefficients*		*Avg. No. of Incorrect*	*Median Non-zero*
				Correct	*Incorrect*	*Non-zero Coef.*	*Coefficients*
Oracle	20	1	1.18	5	0	0	3
		3	10.62	5	0	0	3
		6	42.99	5	0	0	3
	60	1	1.01	5	0	0	3
		3	9.12	5	0	0	3
		6	36.49	5	0	0	3
OLS	20	1	1.64	0	0	5	8
		3	14.79	0	0	5	8
		6	58.58	0	0	5	8
	60	1	1.13	0	0	5	8
		3	10.13	0	0	5	8
		6	40.53	0	0	5	8
Stepwise	20	1	1.46	3.26	0	1.74	4.5
		3	14.13	3.26	0.62	1.74	4
		6	50.84	3.51	1.23	1.48	3
	60	1	1.09	3.9	0	1.10	4
		3	9.89	3.88	0.06	1.12	4
		6	39.92	3.94	0.58	1.06	4
Ridge	20	1	1.61	0	0	5	8
		3	12.38	0	0	5	8
		6	43.15	0	0	5	8
	60	1	1.21	0	0	5	8
		3	10.17	0	0	5	8
		6	39.50	0	0	5	8
LASSO	20	1	1.43	2.06	0	2.94	6
		3	12.16	2.22	0.24	2.78	5
		6	42.87	3.10	0.74	1.9	4
	60	1	1.08	2.34	0	2.66	5.5
		3	9.73	2.26	0	2.74	6
		6	38.99	2.44	0.18	2.56	5
LASSO (lars)	20	1	1.43	2.34	0	2.66	6
		3	12.05	2.56	0.28	2.44	5
		6	43.21	3.42	0.90	1.58	4
	60	1	1.09	2.58	0	2.42	5
		3	9.77	2.58	0	2.42	5
		6	39.07	2.68	0.22	2.32	5
Elastic net	20	1	1.35	1.80	0	3.22	6
		3	11.66	1.84	0.14	3.3	6
		6	41.79	1.84	0.48	3.35	6
	60	1	1.08	2.2	0	2.9	6
		3	9.63	2	0	3.12	6
		6	38.46	1.88	0.08	3.3	6
SCAD	20	1	1.23	3.56	0	1.44	4
		3	12.47	2.76	0.32	2.24	5
		6	44.33	3.48	1	1.52	3
	60	1	1.04	4.22	0	0.78	3
		3	9.64	3.8	0.06	1.2	4
		6	39.40	3.18	0.34	1.82	4
group LASSO	20	1	1.44	1.40	0.42	2.92	7
		3	12.46	1.78	0.42	2.68	6
		6	42.34	3	0.74	2	4
	60	1	1.09	2.26	0.56	2.74	6.5
		3	9.77	1.42	0.56	2.62	6
		6	39.08	1.64	0.58	2.52	6

Table 4.2: Simulation results for Example 1. The true number of zero coefficients is 5 and the non-zero is 3.

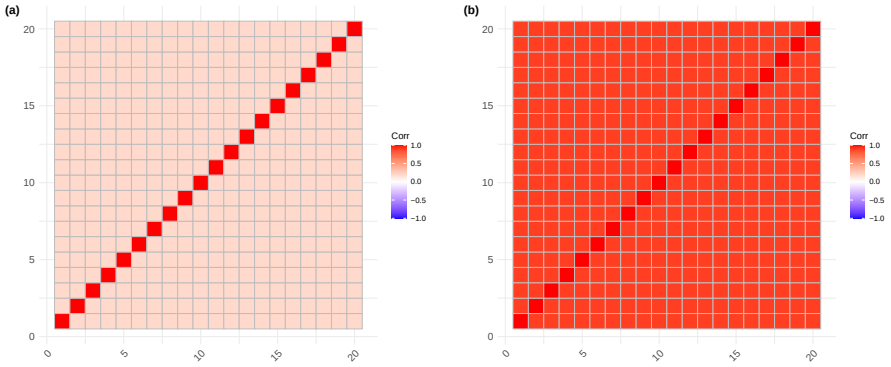


Figure 4.2: Plot of the compound symmetry correlation matrix with (a) $\rho = 0.2$ and (b) $\rho = 0.9$.

Table 4.3 summarize the prediction results. For all methods, there is little difference in the prediction accuracy from increasing the strength of correlation, ρ. This provides evidence that predictions do not suffer from collinearity among the predictor variables. Moreover, the ordinary least squares model has identical MSE for both values of ρ which further supports the conclusion.

For both values of correlation strength ρ, elastic net behaves almost identically to the ridge regression with the optimal value of α optimized to be 0.043 and 0.124, for ρ equals to 0.2 and 0.9 respectively. Both ridge and elastic net performs best compared to the others while stepwise regression and ordinary least squares performs poorly in terms of prediction accuracy.

Method	ρ	*MSE*	*Avg. No. of 0 Coefficients*		*Median Non-zero*
			Correct	*Incorrect*	*Coefficients*
OLS	0.2	13.15	0	0	20
	0.9	13.15	0	0	20
Stepwise	0.2	12.41	0	14.59	5
	0.9	11.58	0	13.44	7
Ridge	0.2	9.62	0	0	20
	0.9	9.09	0	0	20
LASSO	0.2	10.10	0	13.71	6
	0.9	9.38	0	10.04	9.5
LASSO (lars)	0.2	10.13	0	13.86	6
	0.9	9.37	0	10.54	9
Elastic net	0.2	9.56	0	0.97	20
	0.9	9.06	0	0.46	20
SCAD	0.2	10.35	0	16.5	3
	0.9	9.96	0	11.87	8
group LASSO	0.2	10.18	0	13.57	6
	0.9	9.42	0	10.25	10

Table 4.3: Simulation results for Example 2. The true number of zero coefficients is 0 and the non-zero is 20.

In terms of variable selection, ridge regression and elastic net dominate other methods. There is a small number of incorrectly set to zero coefficients when using elastic net and this quantity is zero for ridge regression. This result is expected since the model is defined in such a way that gives advantage to ridge regression while punishing other methods for performing variable selection. All other methods have more than 10 out of 20 predictors incorrectly assigned to zero making those methods having high type II error. Thus, for this example, the simulation results indicate that the ridge regression and elastic net dominate other methods in terms of prediction accuracy and variable selection.

Example 3

For this example, we explore the performance of the variable selection methods when there is non-linear relationship between the covariates. We define the covariates such that the subsequent terms depend on the previous terms by quadratic or higher-order polynomial, and also other function such as exponential. This creates a grouping effect between the covariates. We simulated 50 datasets consisting 60/60/200 observations from the model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\varepsilon, \qquad \varepsilon \sim N(0,1),$$

where the beta coefficients are given by,

$$\boldsymbol{\beta} = (4, 4, 8, 8, -3, -3, \underbrace{0, ..., 0}_{6}, 1, 1)^T.$$

The predictors $\mathbf{X}$ are generated as follows:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1^2, \mathbf{x}_2^2, \mathbf{x}_1^3, \mathbf{x}_2^3, \mathbf{x}_3, \mathbf{x}_4, ..., \mathbf{x}_8, e^{\mathbf{x}_1}, e^{\mathbf{x}_2}].$$

with

$$\mathbf{x}_i = Z_1 + \varepsilon_i^x, \quad Z_1 \sim N(0,1), \quad i = 1, 2,$$
$$\mathbf{x}_i \sim N(0,1), \quad \mathbf{x}_i \text{ independent and identically distributed}, \quad i = 7, ..., 12,$$

where ε_i^x are independent and identically distributed $N(0, 0.01)$, for $i = 1, 2$. The correlation is not specified but instead exists from the way we define the model. Figure 4.3 shows the corresponding correlation plot.

The effect of noise is varied by considering $\sigma \in \{1, 3\}$ with corresponding signal-to noise ratio 204.375 and 22.708 respectively. The true model is given by

$$f(X) = X_1\beta_1 + X_2\beta_2 + X_1^2\beta_3 + X_2^2\beta_4 + X_1^3\beta_5 + X_2^3\beta_6 + \exp(X_1)\beta_7 + \exp(X_2)\beta_8.$$

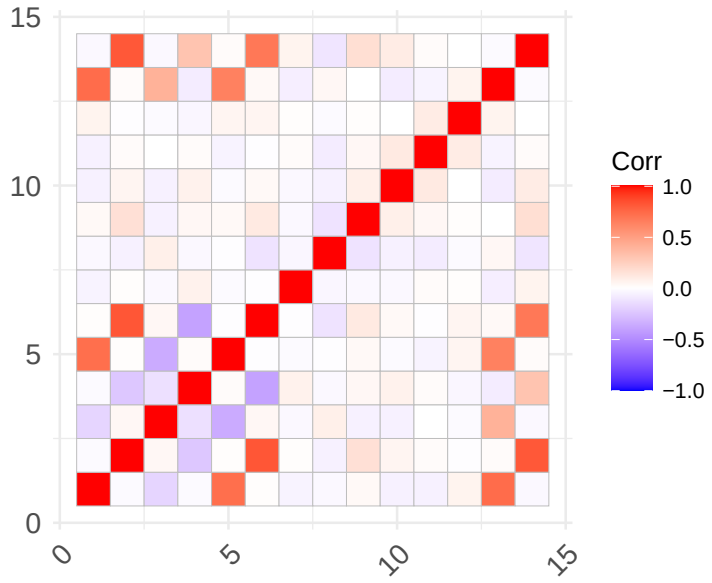


Figure 4.3: Plot of correlation matrix for predictors $\mathbf{X}$.

Table 4.4 shows the result of the simulation. For all methods, the median MSE decreases significantly as we decrease the noise level. Although we would expect elastic net or group LASSO to dominate in terms of prediction accuracy, SCAD appears to outperform other procedures under all considered conditions. Group LASSO, elastic net and LASSO performs considerably well while ridge regression, stepwise regression and also ordinary least squares fall behind by a large difference. It is also important to note that the oracle procedure also performs poorly compared to other penalized least square methods. This may due to the definition of model which makes ordinary least squares a bad fit in general even when the true number of non-zero predictors are used to fit the oracle.

Stepwise regression and SCAD both performs quite well in selecting the variables. Stepwise regression correctly assigned the most number of zero-coefficients as well as the smallest number of incorrect non-zero coefficients. This shows that stepwise regression is still useful in this type of model if our objective is mainly to select variable rather than making accurate predictions. Overall, we can conclude that SCAD performs better than other methods in terms of prediction accuracy and still did a great job in selecting variables.

Example 4

In this example, we grouped the observations by specifying the correlation for each non-overlapping groups. We simulated 50 datasets consisting of 60/60/200 observations from the linear model (4.2). We consider large number but small to moderate size of predictors defined as follows,

$$\boldsymbol{\beta} = (\underbrace{2, ..., 2}_{5}, \underbrace{3, ..., 3}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{0.4, ..., 0.4}_{5})^T. \tag{4.4}$$

Method	σ	*MSE*	*Avg. No. of 0 Coefficients*		*Avg. No. of Incorrect*	*Median Non-zero*
			Correct	*Incorrect*	*Non-zero Coef.*	*Coefficients*
Oracle	1	3.43	6	0	0	8
	3	30.85	6	0	0	8
OLS	1	4.48	0	0	6	14
	3	40.33	0	0	6	14
Stepwise	1	5.40	4.38	1.26	1.62	8
	3	25.04	4.52	1.86	1.48	7
Ridge	1	20.68	0	0	6	14
	3	31.85	0	0	6	14
LASSO	1	1.70	1.62	0.52	4.38	12
	3	12.13	0.8	0.6	5.2	13
LASSO (lars)	1	1.44	0.74	0.36	5.26	13
	3	12.19	0.92	0.78	5.08	13
Elastic net	1	1.49	0.84	0.12	5.24	13
	3	11.98	0.54	0.38	5.5	13
SCAD	1	1.37	3.8	0.92	2.2	9
	3	11.81	3.24	1.22	2.76	9
group LASSO	1	1.93	0.64	0.54	5.2	13
	3	12.62	0.7	1.02	5.3	13

Table 4.4: Simulation results for Example 3. The true number of zero coefficients is 6 and the non-zero is 8.

The predictors $\mathbf{X} \sim N(\mathbf{0}, \mathbf{\Sigma})$ are grouped by compound symmetry correlation structure given below,

$$\text{corr}(X_i, X_j) = \begin{cases} 1 & i = j, \\ 0.5 & i \neq j,\ \forall i, j = 1, ..., 5, \\ 0.9 & i \neq j,\ \forall i, j = 6, ..., 10, \\ 0.2 & i \neq j,\ \forall i, j = 16, ..., 20, \\ 0 & \text{elsewhere}, \end{cases}$$

as plotted in Figure 4.4.

We allow the effect of noise to vary by considering $\sigma \in \{1, 3\}$ as in previous examples with corresponding signal-to-noise ratio of 127.21 and 14.13 respectively. The true model is given by

$$f(X) = X_1\beta_1 + X_2\beta_2 + ... + X_{10}\beta_{10} + X_{16}\beta_{16} + X_{17}\beta_{17} + ... + X_{20}\beta_{20}.$$

The simulation results are summarized in Table 4.5. From Table 4.5, we can observe that elastic net has the best prediction accuracy and performs similar to the oracle procedure when the noise level is low, owing to the grouping property. As we increase the noise level, elastic net surprisingly dominates the oracle procedure in terms of prediction accuracy. Ridge regression, LASSO and group LASSO performs quite well while SCAD suffers from relatively higher MSE at both noise level. The ordinary least squares and stepwise regression performs slightly worse than other methods.

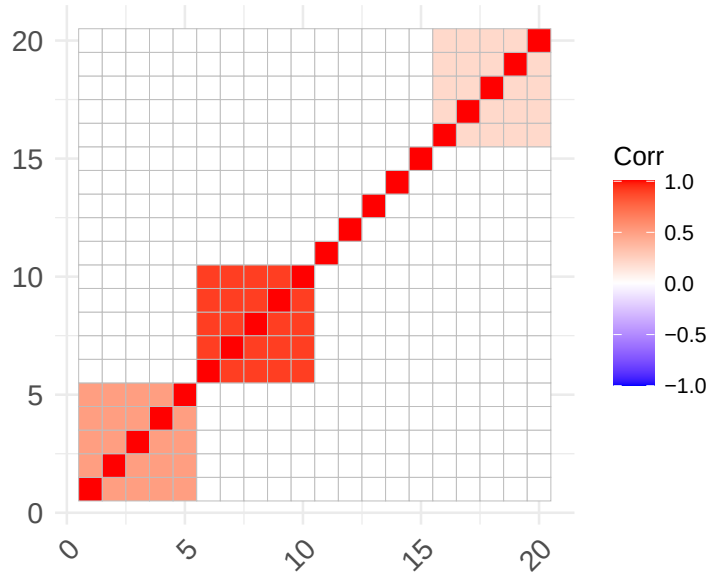


Figure 4.4: Plot of the specified correlation matrix.

Despite having the lowest MSE, elastic net tends to select more variables than required. Notice that from Table 4.5, elastic net has the highest type II error but the lowest type I error. Stepwise regression performs quite well in terms of variable selection when the noise level is low. LASSO, SCAD, and group LASSO performs well in selecting variables when the noise level is high but similar to elastic net, tend to select more variables than needed. For this example, we chose LASSO as the best method since its prediction accuracy is not far behind elastic net and the oracle procedure but also performs extremely well in variable selection.

Example 5

In this example, we simulated 100 datasets consisting of 60/60/200 observations from the group model

$$\mathbf{y} = \sum_{\ell=1}^{4} \mathbf{X}_\ell \boldsymbol{\beta}_\ell + \sigma\varepsilon, \qquad \varepsilon \sim N(0,1),$$

where the 320×20 design matrix, $\mathbf{X}$, follows standard normal. We consider large number with small to moderate sized beta parameters defined as,

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4)^T = (\underbrace{0.5, ..., 0.5}_{5}, \underbrace{2, ..., 2}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{5, ..., 5}_{5})^T.$$

In this example, we did not specify the correlation between predictors. The grouping effect is based on the group model we used. The effect of noise is varied by considering $\sigma \in \{1, 3\}$ with corresponding signal-to-noise ratio 91.94 and 10.21 respectively. The true model used to fit the oracle procedure is given by,

$$f(X) = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \mathbf{X}_4\boldsymbol{\beta}_4$$

Method	σ	*MSE*	*Avg. No. of 0 Coefficients*		*Avg. No. of Incorrect*	*Median Non-zero*
			Correct	*Incorrect*	*Non-zero Coef.*	*Coefficients*
Oracle	1	1.30	5	0	0	15
	3	11.66	5	0	0	15
OLS	1	1.53	0	0	5	20
	3	13.81	0	0	5	20
Stepwise	1	1.48	3.78	0.56	1.22	16
	3	13.73	3.74	3.66	1.26	13
Ridge	1	1.40	0	0	5	20
	3	11.30	0	0	5	20
LASSO	1	1.36	2.26	0.06	2.74	18
	3	11.68	2.74	1.82	2.26	15
LASSO (lars)	1	1.37	2.46	0.08	2.54	17
	3	11.71	2.9	1.98	1.37	15
Elastic net	1	1.32	1.62	0.04	3.36	19
	3	10.98	1.38	0.92	3.48	18.5
SCAD	1	1.40	2.94	0.22	2.06	17
	3	12.86	2.06	2.34	2.94	16
group LASSO	1	1.38	2.1	0.06	2.9	18
	3	11.77	2.56	1.74	2.44	15

Table 4.5: Simulation results for Example 4. The true number of zero coefficients is 5 and the non-zero is 15.

The simulation results are summarized in Table 4.6. From Table 4.6, we can see that the SCAD performs better than the oracle when the noise level is high and almost as good as the oracle when we reduce the noise. Group LASSO comes second after SCAD in terms of prediction accuracy which is expected. Ridge, stepwise regression and ordinary least squares performs poorly at high noise level. Stepwise regression and ordinary least squares predictions improve when we reduce the noise level. Ridge regression remains with high MSE.

In terms of variable selection, the SCAD still performs well compared to other methods. However, the best method to select variables would be the stepwise regression as it has the most predictors correctly assigned to zero, the lowest type II error but slightly higher type I error. Group LASSO fails to select the grouped variables at both noise level. LASSO and elastic net performs moderately compared to the others. Thus, it is apparent that SCAD still dominates even in this example where grouped models is used.

From the result, there was no compelling evidence which suggest the simulated groups improves the variable selection. Similar result is also seen in Ogutu and Piepho (2014). Based on the paper, one possible reason for this is the simulated group does not have strong within- group correlation or possibly have strong between- group correlation. This causes group LASSO to select more variables than needed. In practice, methods such as K-means clustering and K-means spatial clustering can be used to identify the underlying grouping structure depending on the complexity of the data (Ogutu and Piepho, 2014).

Method	σ	*MSE*	*Avg. No. of 0 Coefficients*		*Avg. No. of Incorrect*	*Median Non-zero*
			Correct	*Incorrect*	*Non-zero Coef.*	*Coefficients*
Oracle	1	1.35	5	0	0	15
	3	12.15	5	0	0	15
OLS	1	1.53	0	0	5	20
	3	13.75	0	0	5	20
Stepwise	1	1.48	3.84	0.2	1.16	16
	3	13.12	3.82	2.7	1.18	14
Ridge	1	2.30	0	0	5	20
	3	13.47	0	0	5	20
LASSO	1	1.43	1.08	0.02	3.92	19
	3	12.39	1.2	0.76	3.8	18
LASSO (lars)	1	1.43	0.98	0.02	4.02	20
	3	12.41	1.52	0.96	3.48	17.5
Elastic net	1	1.43	0.84	0.02	4.1	19
	3	12.37	1.1	0.6	3.98	18
SCAD	1	1.41	2.5	0.06	2.5	17
	3	12.03	2.82	1.76	2.18	15
group LASSO	1	1.42	0.2	0	4.34	20
	3	12.18	0.4	0.2	4.6	20

Table 4.6: Simulation results for Example 5. The true number of zero coefficients is 5 and the non-zero is 15.

From this chapter, we can see that, in general, the SCAD shows an excellent performance both in terms of accuracy of prediction and selecting important variables, especially in low noise settings. Other methods tend to specialize in different settings. For instance, when all the true parameters are significant, ridge regression is favoured as it does not performs variable selection. Similarly, in the group settings, we should expect elastic net and group LASSO to perform best. However, our simulation result shows that the SCAD performs best in two out of three grouped settings that we considered for the potential reasons discussed above. We will next assess how these methods perform when dealing with real data in Chapter 6.

Chapter 5

Application - Diabetes data

5.1 Data

The data for this example is obtained from "Least Angle Regression" by Efron et al. (2004) and can be accessed through the GitHub link provided in Appendix A. The scaled version is also available in the R package `lars`. The data is based on a study conducted on 442 diabetes patients, measuring ten baseline variables; age, sex, body mass index (bmi), average blood pressure or more formally, the mean arterial pressure (map), and six blood serum measurements (tc, ldl, hdl, tch, ltg, glu) as well as the response of interest. These variables are the predictors in a regression model with the quantitative measure of disease progression one year after baseline as the response variable. For the analysis, we consider the variable sex (1,2) as quantitative since the standardized coefficient estimates will be the same as if the variable being treated as binary data (0,1) or categorical data. This assumption will only affect the intercept term which we omit throughout the analysis. The model can be useful in determining which factors promote the progression of the disease as well as predicting disease progression for future patients using their baseline measurements.

The predictors, $\mathbf{X}$ are standardized and the summary statistics of the data is given in Table 5.1. The correlation matrix for the Diabetes dataset is presented visually in Figure 5.2. From Figure 5.2, we observe some of the blood serum measurements appear to have moderate to high correlation with each other. In particular, the variables tc, ldl, hdl, tch and ltg form a group of variables with high pairwise correlation with substantial correlation between ldl and tc, as well as hdl and tch. The remaining blood serum measurement, glu has moderate correlation with all blood serum measurements. Other correlations ranges between low to moderate. Thus, we can conclude that collinearity exists within this dataset.

Variables	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
age	-2.2517	-0.7833	0.1130	0	0.7996	2.3253
sex	-0.9375	-0.9375	-0.9375	0	1.0643	1.0643
bmi	-1.8958	-0.7188	-0.1530	0	0.6562	3.5817
map	-2.3604	-0.7698	-0.1191	0	0.7485	2.7729
tc	-2.6624	-0.7192	-0.0907	0	0.5955	3.2322
ldl	-2.4279	-0.6375	-0.0802	0	0.6267	4.1745
hdl	-2.1484	-0.7375	-0.1383	0	0.6155	3.8048
tch	-1.6043	-0.8294	-0.0544	0	0.7205	3.8899
ltg	-2.6480	-0.6982	-0.0409	0	0.6811	2.8055
glu	-2.8931	-0.6968	-0.0226	0	0.5863	2.8479
y	25	87	140.5	152.1	211.5	346

Table 5.1: Summary statistics of the standardized covariates for the diabetes data set.

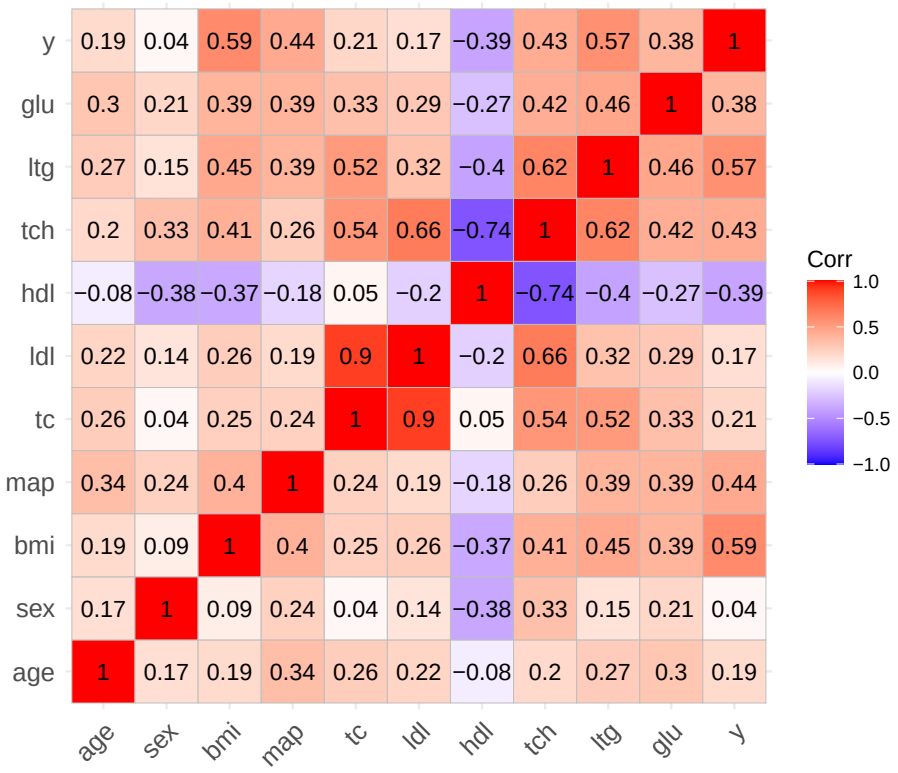


Figure 5.1: Correlation plot for the diabetes data set.

5.2 Estimation, Model Selection and Prediction

The data is split roughly into $75 - 25\%$ ratio of training to test size which corresponds to 332 and 110 observations respectively, using random sampling without replacement. Given the data is low-dimensional, fitting the ordinary least squares would not lead to overfitting. Thus, ordinary least squares will also be fitted on the training data together with all the methods discussed in Chapter 3, which are stepwise regression, ridge regression, LASSO, elastic net, SCAD and also group LASSO.

Given we only a have a single data set, model selection is performed by resampling data from the training set using 10-fold cross-validation, simultaneously optimizing tuning parameters where relevant. The tuning parameter which yield the lowest cross-validation error is selected and used to predict responses based on the test set. The performance of each model is then assessed based on the computed test error.

5.3 Results

We begin by fitting the ordinary least squares. The model we are fitting is

$$f(X) = X_{age}\beta_1 + X_{sex}\beta_2 + X_{bmi}\beta_3 + X_{map}\beta_4 + X_{tc}\beta_5 + X_{ldl}\beta_6 + X_{hdl}\beta_7 + X_{tch}\beta_8 + X_{ltg}\beta_9 + X_{glu}\beta_{10}.$$

The standardized coefficients obtained from the fit together with the corresponding standard error and p-value are shown in Table 5.2

Methods	Estimate	Std. Error	t value	p - value	VIF
age	-0.9707	3.3153	-0.293	0.7699	1.2071
sex	-10.2593	3.3508	-3.062	0.0024	1.2982
bmi	27.8906	3.6534	7.634	2.62×10^{-13}	1.5337
map	16.1286	3.5082	4.597	6.16×10^{-6}	1.4539
tc	-51.3677	21.2534	-2.417	0.0162	46.8037
ldl	31.8564	17.0869	1.864	0.0632	31.2802
hdl	14.2458	10.7827	1.321	0.1874	14.4312
tch	15.7318	8.4907	1.853	0.0648	8.8066
ltg	38.3529	9.0778	4.225	3.12×10^{-5}	9.7394
glu	3.7395	3.6474	1.025	0.3060	1.5268

Table 5.2: Standardized coefficients together with the corresponding standard error, p-value and variance inflation factor (VIF).

At 5% significance level, we have five variables which are significant in predicting the measure of disease progression which are sex, bmi, map, tc, and ltg. Next, we check the multicollinearity in the dataset. Apart from inspecting the correlation matrix, we use the *variance inflation factor* (VIF) which measures the ratio of the variance of $\hat{\beta}_j^o$ when fitting the full model divided variance of $\hat{\beta}_j^o$ if fit on its own (James et al., 2014). The computed VIF for each beta coefficient are summarized in the last column of Table 5.2. As a general rule, a VIF

value which exceeds 5 indicates a substantial collinearity exists. Therefore, the result supports our initial observation in which collinearity exists between the variables tc, ldl, hdl, tch and ltg.

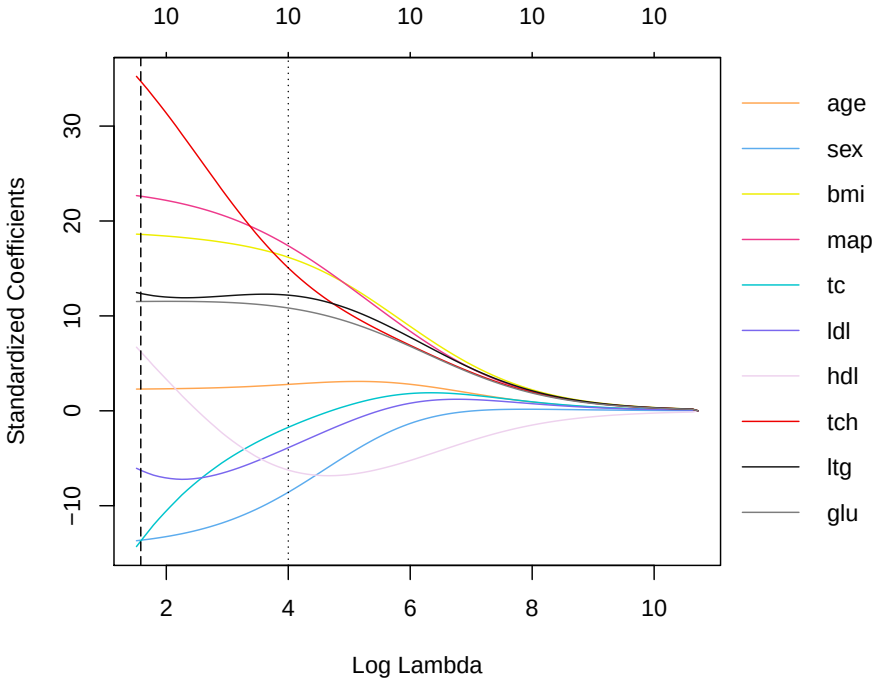
We proceed by fitting other models using stepwise regression, ridge regression, LASSO, elastic net, SCAD, and group LASSO. The CV error curves are shown in part (b) of Figure 5.2 to Figure 5.6 which correspond to ridge regression, LASSO, elastic net, SCAD, and group LASSO. The value of λ, denoted by λ_{min}, which yield the minimum CV error is indicated by the dashed black line while λ, denoted by λ_{1se}, which gives the most regularized model such that the cross-validated error are within one standard error from the minimum is labelled using the dotted black line. The latter is a built-in feature in `glmnet` package, hence the dotted line is only shown in the ridge, LASSO and elastic net plots. As mentioned before, these values of λ are selected to fit the regression model for each corresponding method. Note that from all the plots, the variance increases as λ increases which suggest it could be beneficial to use 5-fold CV instead.

Table 5.3 shows the value of tuning parameters selected using 10-fold CV and AIC for each method. Recall from Chapter 2, the performance of SCAD using $a = 3.7$ works similarly to the value chosen by generalized cross-validation (GCV) method (Fan and Li, 2001). Thus the value of a is fixed to be 3.7 when fitting the model for diabetes dataset.

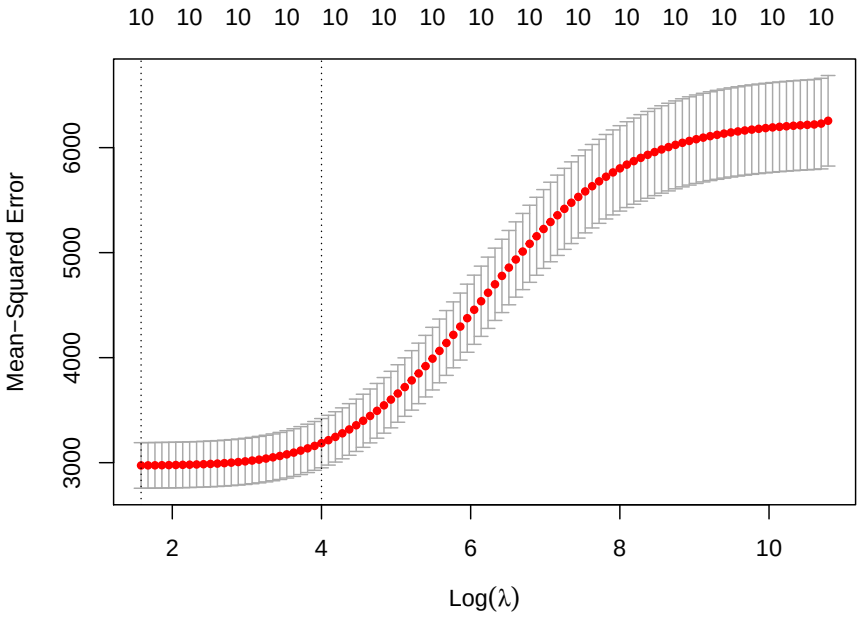
Method	Primary			Secondary		
	Type	CV	AIC	Type	CV	Fixed
Stepwise	p		6			
Ridge	λ	4.862				
LASSO	λ	0.201				
Elastic Net	λ	0.546		α	0.1	
SCAD	λ	-0.721		a		3.7
group LASSO	λ	-1.605				

Table 5.3: Tuning parameters selected for the diabetes data.

The coefficient profiles for the ridge regression, LASSO, elastic net, SCAD and group LASSO are shown in part (a) of Figure 5.2 to Figure 5.6. On top of Figure 5.2, 5.3, 5.4 and part (b) of Figure 5.5, and Figure 5.6 are the numbers of non-zero regression coefficients. From the plots, we can observe the shrinkage effect of the tuning parameters. As expected, none of the coefficients are being set to zero for ridge regression. For LASSO, elastic net, SCAD and group LASSO, increasing the value of λ leads to shrinkage of the regression coefficients and some of these even become zero. For ridge regression, the LASSO, tch and tc get shrunk the most while for SCAD and group LASSO apply the most shrinkage to tc and ltg. The coefficient paths for LASSO, elastic net and group LASSO parameters are similar. We can infer that the performance of LASSO is comparable to the elastic net since the parameters are shrunk to zero at a similar rate.

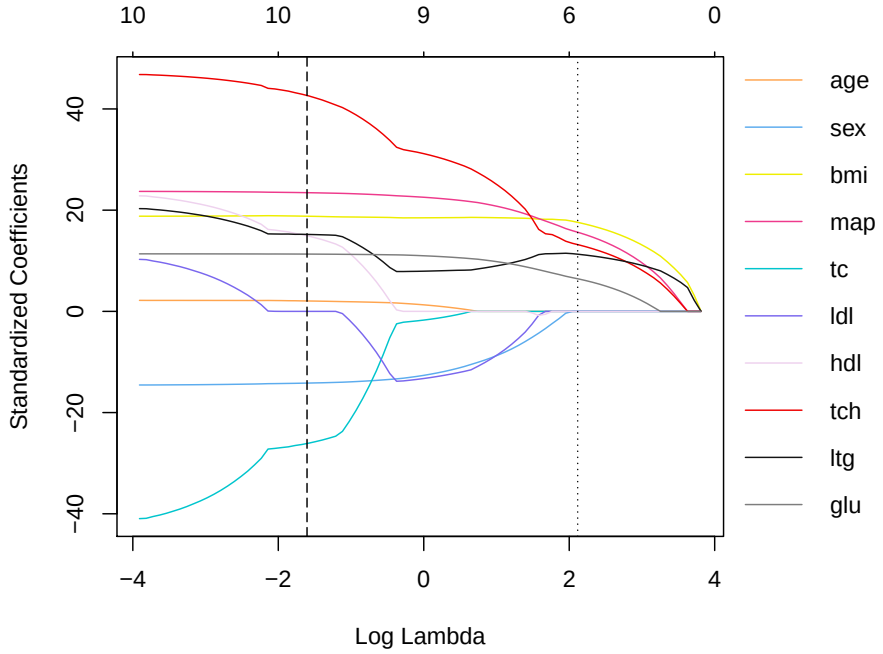


(a) Coefficient profiles

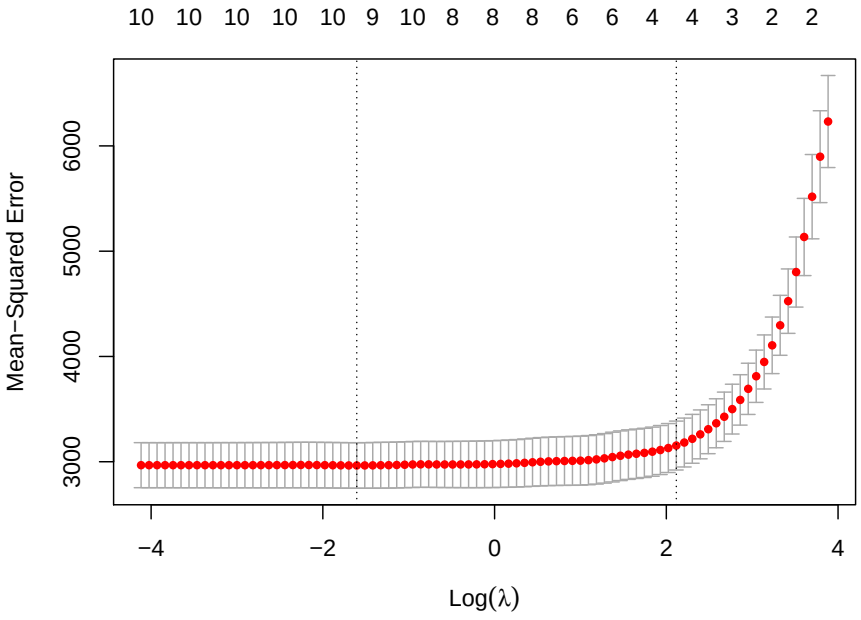


(b) CV curves

Figure 5.2: Coefficient profiles and the CV curves for ridge regression. Dashed line on the left (- -) corresponds to lambda which gives minimum mean cross-validated error and dotted line on the right (. . .) correspond to lambda that gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

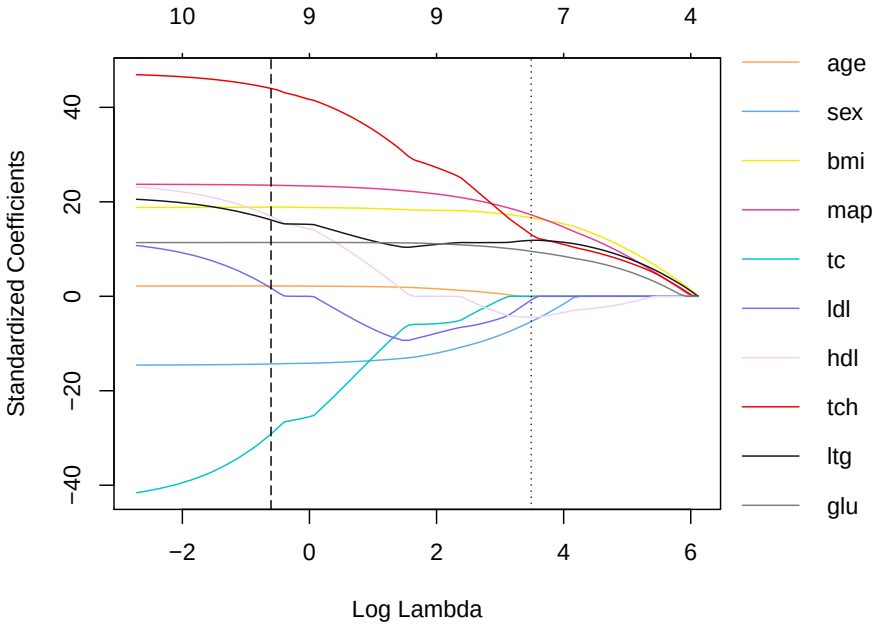


(a) Coefficient profiles

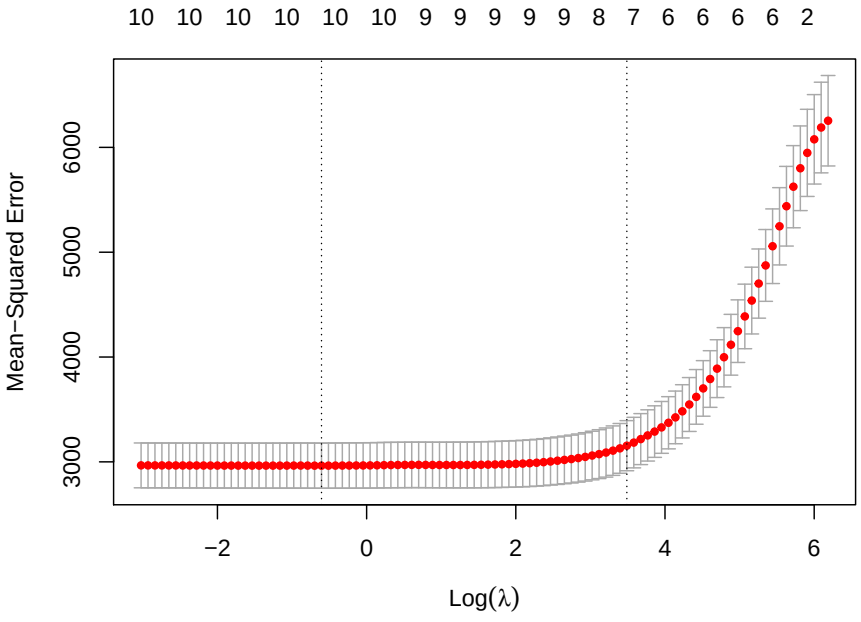


(b) CV curves

Figure 5.3: Coefficient profiles and the CV curves for LASSO. Dashed line on the left (- -) corresponds to lambda which gives minimum mean cross-validated error and dotted line on the right (. . .) correspond to lambda that gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

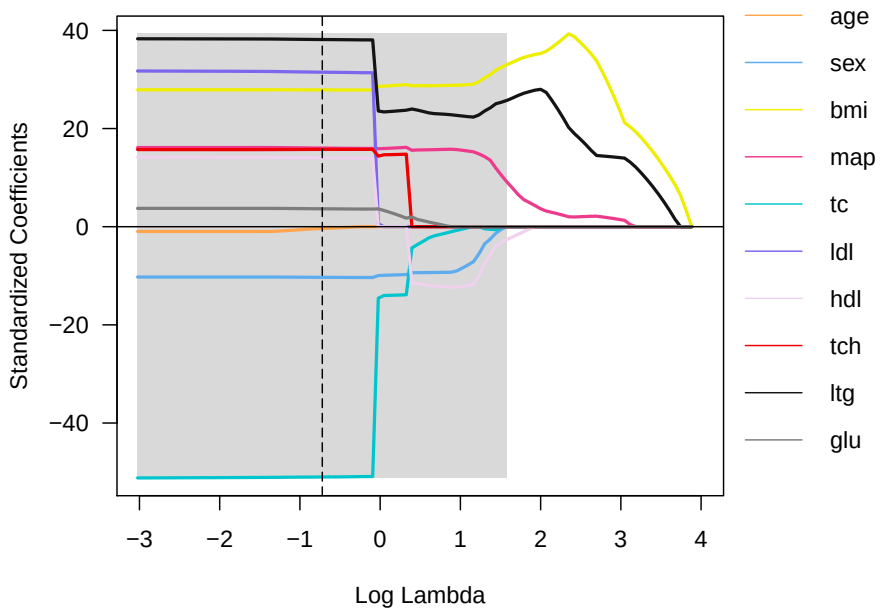


(a) Coefficient profiles

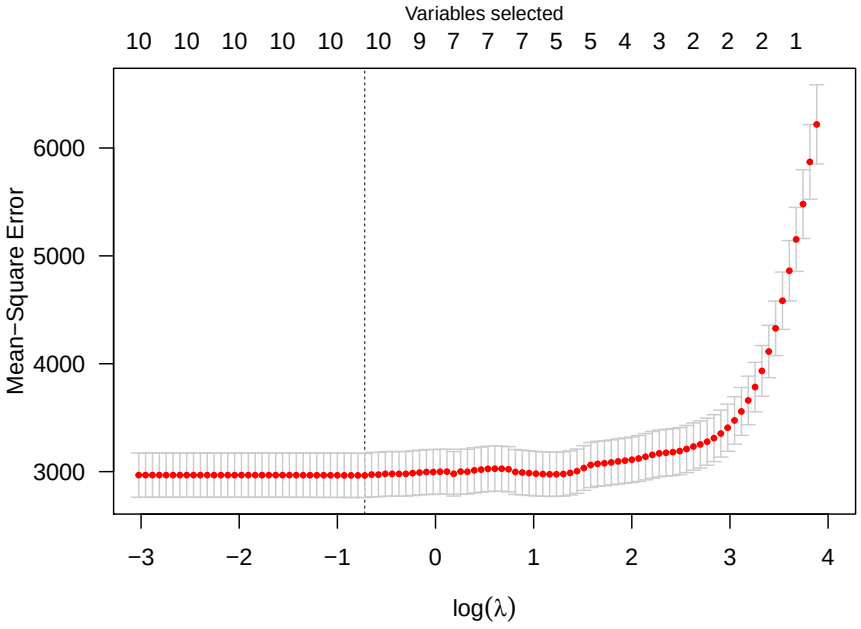


(b) CV curves

Figure 5.4: Coefficient profiles and the CV curves for elastic net. Dashed line on the left (- -) corresponds to lambda which gives minimum mean cross-validated error and dotted line on the right (. . .) correspond to lambda that gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

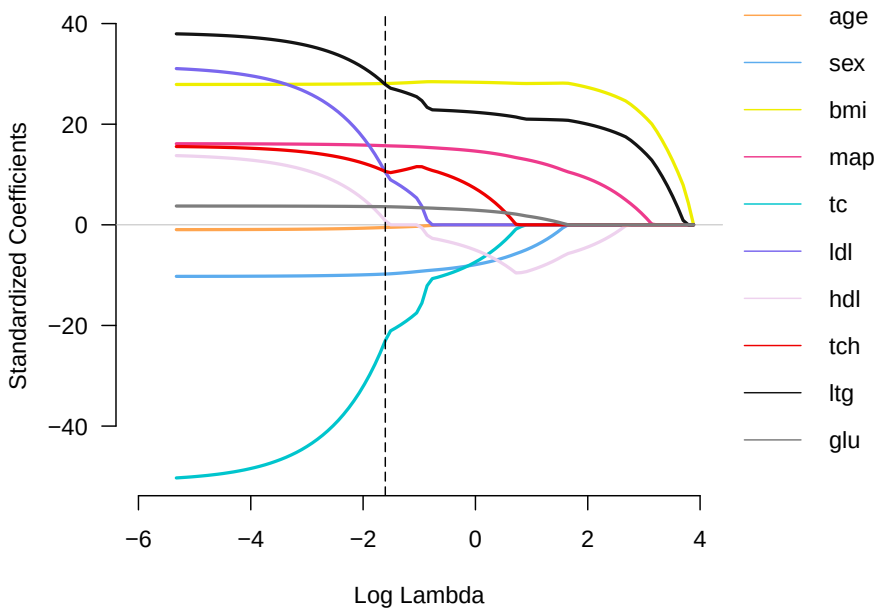


(a) Coefficient profiles

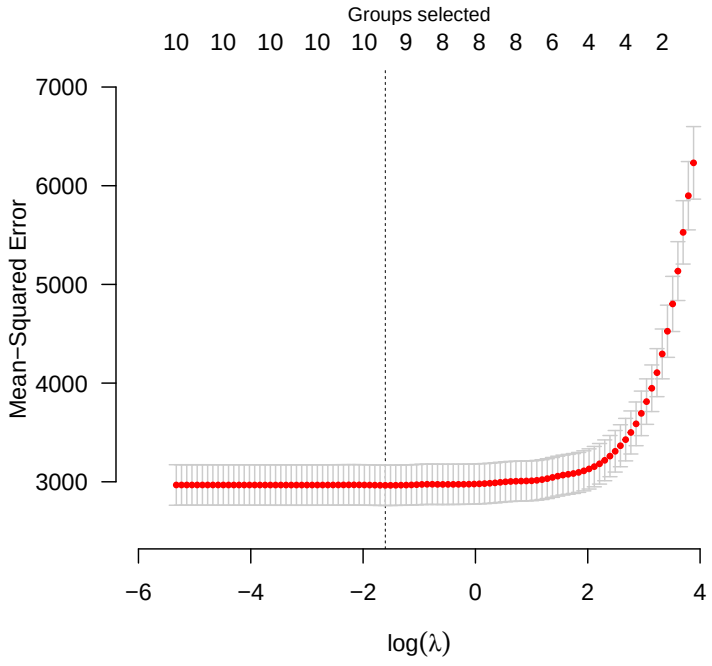


(b) CV curves

Figure 5.5: Coefficient profiles and the CV curves for SCAD. Dashed line (- -) corresponds to lambda which gives minimum mean cross-validated error.



(a) Coefficient profiles



(b) CV curves

Figure 5.6: Coefficient profiles and the CV curves for group LASSO. Dashed line (- -) corresponds to lambda which gives minimum mean cross-validated error.

The standardized coefficients for all the methods discussed are shown in Table 5.4 and 5.5. Recall the five variables found significant under the model fitted using the ordinary least squares are sex, bmi, map, tc and ltg. Based on Table 5.4 and 5.5, all models, except LASSO and elastic net using λ_{1se}, include these five variables. From the results, we can see that stepwise regression, as well as LASSO and elastic net, especially when fitted using λ_{1se}, have more tendency of producing a sparse model.

The variable age has a relatively lower coefficient values for all the methods. Stepwise regression, LASSO and elastic net set the coefficient for age to zero. Thus, based on the result, we can infer that age does not affect disease progression. From Table 5.5, we can also argue that LASSO and elastic net model obtained using λ_{1se} are likely to be wrong in setting the coefficient for variable tc to zero as other methods have high estimated value for the corresponding variable. The variable ldl was shrunk by all methods except SCAD. The variable hdl has a lower coefficient estimates set by all methods including the ordinary least squares. Five variables which have a significantly higher estimates are bmi, map, tc, tch and ltg.

The test error computed based on the predicted values made by each method are shown in the last row of Table 5.4 and 5.5. To better compare the test error of each method, we create a simple bar plot for all the methods with least squares being the baseline as shown in Figure 5.7. Note that for ridge regression, LASSO and elastic net, we only include the test error computed based on the model fitted using λ_{min} as this shows a better prediction accuracy, as opposed to using λ_{1se}. Based on Figure 5.7, all of the methods perform relatively well compared to the least squares with SCAD being the closest to the least squares method. This shows that SCAD actually performs the worst among all the discussed methods for this particular data set. Based on Figure 5.7, it is apparent that ridge regression has the lowest test error while other methods seem to have similar resulting test error.

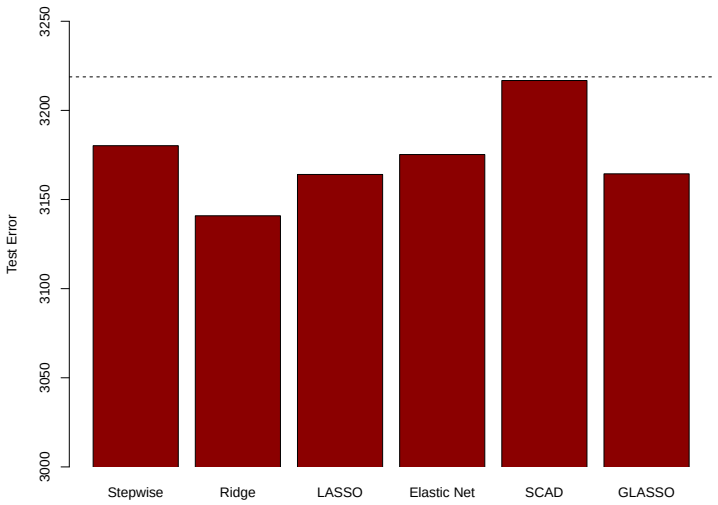


Figure 5.7: Test Error computed based on the predicted values from each methods. The horizontal dotted line represents the test error for the least squares method.

	OLS	Stepwise	SCAD	Group LASSO
age	-0.97	0	-0.61	-0.52
sex	-10.26	-9.58	-10.29	-9.78
bmi	27.89	29.27	27.88	28.09
map	16.13	16.44	16.06	15.72
tc	-51.37	-13.69	-50.99	-23.01
ldl	31.86	0	31.51	10.46
hdl	14.25	0	14.06	0.82
tch	15.73	14.91	15.73	10.61
ltg	38.35	24.13	38.16	27.89
glu	3.74	0	3.69	3.60
Test Err.	3218.80	3180.20	3216.76	3164.41

Table 5.4: Standardized coefficients selected based on CV (for SCAD, group LASSO) and AIC (for stepwise regression).

	Ridge		LASSO		Elastic net	
	min.	1se	min.	1se	min.	1se
age	2.29	2.79	2.06	0	2.19	0
sex	-13.64	-8.59	-14.17	0	-14.33	-5.38
bmi	18.59	16.17	18.80	17.59	18.87	16.63
map	22.61	17.39	23.45	15.64	23.51	17.24
tc	-13.73	-1.74	-26.11	0	-29.14	0
ldl	-6.26	-3.89	0	0	1.74	-0.79
hdl	6.22	-6.24	14.95	0	16.91	-4.48
tch	34.72	15.07	42.66	13.20	44.00	13.08
ltg	12.36	12.19	15.21	11.30	16.16	11.83
glu	11.52	10.83	11.32	6.44	11.38	9.52
Test Error	3140.88	3222.77	3164.05	3246.90	3175.24	3244.33

Table 5.5: Standardized coefficients selected based on CV.

Recall that one of our objective is to use this data to predict disease progression for future patients. Thus, we would prefer a model that can give the most accurate prediction. Based on the analysis, we choose ridge regression as the best method for this data set. This is because it yields the lowest test error which helps to achieve our aim. Plus, there is no clear evidence of sparse model are produced by other methods, unless fitted using λ_{1se} which results in poor prediction performance. The LASSO only sets the coefficient of a single variable, ldl to zero which again, does not provide substantial proof that sparse model is the most appropriate model to fit for this data set. Stepwise regression produce a sparse model but the computed test error is relatively higher which makes it less favourable compared to ridge regression. Since the true parameters are unknown, we may infer from the fact that ridge regression performs best, that all the covariates are significant in predicting the measure of disease progression.

Chapter 6

Discussion

6.1 Summary

In this report, we have studied modern shrinkage methods that extend the ordinary least squares. These methods help to overcome the limitation of least squares method to produce a sparse model which ease interpretation and also provide accurate predictions. Table 6.1 provides the key features for the methods discussed in Chapter 2. In the linear settings, we might prefer SCAD over other methods as it possesses the oracle property, allowing it to identify the significant covariates in the true model. On top of that, SCAD is also applicable in high-dimensional settings which makes SCAD a better option compared to adaptive LASSO.

On the other hand, elastic net and group LASSO are preferred when grouping exists between covariates. Under proper group assignment (refer Chapter 4), we would expect group LASSO and elastic net to show excellent performance, both in terms of prediction accuracy and variable selection.

Method	Analytical solution	Variable selection	Applicable in high-dimension	Grouping effect	Oracle
Best-subset sel.	yes	yes	no	no	no
Stepwise	yes	yes	yes	no	no
Ridge	yes	no	yes	no	no
LASSO	no	yes	no	no	no
BRidge	no	yes	yes	no	no
NNG	no	yes	no	no	no
SCAD	no	yes	yes	no	yes
Ada. LASSO	no	yes	no	no	yes
Elastic net	no	yes	yes	yes	no
Group LASSO	no	yes	yes	yes	no

Table 6.1: Summary of methods discussed in Chapter 2.

However, an important thing to note is that there is not a single method that always performs better than all other methods under all conditions. Instead, the performance of these methods are highly data-dependent as shown in Chapter 4 and 5 of this report. In example 1, we see that elastic net performs best when when the noise level is high. As we decrease the noise level, SCAD dominates in terms of prediction accuracy. In example 2, we create a settings which favour the ridge regression with non of the true coefficients are equal to zero. Under this setting, SCAD falls behind compared to ridge and elastic net. Example 3 presents the grouping effect by creating linear model with polynomial terms and also exponential terms. This creates high correlation between the covariates. Elastic net falls slightly behind SCAD. Similar result can be seen for Example 4 and 5. From our simulation studies, we see that both elastic net and SCAD are consistent in giving the best prediction accuracy and selecting significant variables.

Given the theory in Chapter 2 and simulation results in Chapter 4, we then present a comparative study between the methods using the diabetes data set in Chapter 5. The data is found to be correlated and our result shows that ridge regression performs the best. As opposed to its consistent excellent performance in simulation studies, the SCAD actually performs the worst for this particular data set. This supports the fact that no particular method surpasses the others in every scenario.

6.2 Conclusion

Regression models have an abundance of applications in our digital society. There are still much to explore in the area of model selection. As more data are becoming available, many different regularization methods have been introduced to address any particular problem, making no method consistently dominates the others.

Our report has presented the background theory behind some of the available methods with particular focus on modern shrinkage methods, together with some evidence of their practicality through a number of simulation studies and analysis of real data. Since we only cover cases where $p < n$, further exploration on the application of these methods in high-dimensional, where feasible, is still needed. Other possible improvements would be to explore other variable selection methods such as minimax concave penalty (MCP) (Zhang, 2010), sure independence screening (SIS) (Fan and Lv, 2008), and potentially other regression models namely additive model, partial linear model and also non-parametric model.

Appendix A

R codes

R Code to produce the penalty plots in Chapter 2

R code to produce Figure 2.1.

```
### SET-UP THE DATA ###
# lambda = 3
x = seq(from = -20, to = 20, length.out = 1000)
y1 = 1/(1+3)*x
y2 = sign(x)*pmax((abs(x)-3),0)
# set-up the plot window
par(mar = c(6, 6, 3, 3))
par(cex.axis=1.5, cex.main=1.3, cex.lab = 1.3)
par(mfrow=c(3,1))

### HARD THRESHOLDING / SUBSET-SELECTION ###
y4 = x*1*(abs(x)>3)
plot(x,y4,type='l',xlim = c(-10,10), ylim = c(-10,10),
     xaxs="i", yaxs="i",  ylab = expression(hat(beta)^SS),
     xlab = expression(hat(beta)^o),font.main = 1,
     main = "(a)")
lines(x, x, type = "l",lty=3, lwd = 0.8)

### RIDGE REGRESSION ###
plot(x,y1,type='l', xlim= c(-20,20), ylim= c(-20,20),
        xaxs="i", yaxs="i",
        ylab = expression(hat(beta)^R),
        xlab = expression(hat(beta)^o),
        font.main = 1,main = "(b)")
lines(x, x, type = "l",lty=3, lwd = 0.8)

### LASSO ###
plot(x,y2,type='l', xlim= c(-20,20), ylim= c(-20,20),
        xaxs="i", yaxs="i",
        ylab = expression(hat(beta)^L),
        xlab = expression(hat(beta)^o),
        font.main = 1,  main = "(c)")
lines(x, x, type = "l",lty=3, lwd = 0.8)
```

R code to produce Figure 2.2.

```
par(mar = c(4, 5, 4, 4))
par(cex.axis=1.5, cex.main=1.5)
par(mfrow=c(1,2))
### BRIDGE WITH q=3 ###
x = seq(from = -20, to = 20, length.out = 1000)
y = seq(from = -20, to = 20, length.out = 1000)
z = outer(x,y,function(x,y) x -6*(abs(y)**2)*sign(y)/2 -y)
contour(x,y,z,levels=0, drawlabels = FALSE,
        ylab = expression(hat(beta)^B),
        xlab = expression(hat(beta)^o),
        xlim= c(-2,2), ylim= c(-2,2),
        xaxs="i", yaxs="i",font.main = 1,
        main = "(a)")
lines(x, x, type = "l",lty=3)
### BRIDGE WITH q=1.5 ###
x = seq(from = -20, to = 20, length.out = 1000)
y = seq(from = -20, to = 20, length.out = 1000)
z = outer(x,y,function(x,y) x -3*(abs(y)**0.5)*sign(y)/2 -y)
contour(x,y,z,levels=0, drawlabels = FALSE, xlim= c(-2,2),
        ylab = expression(hat(beta)^B),
        xlab = expression(hat(beta)^o),
        ylim= c(-2,2), xaxs="i",
        yaxs="i", font.main = 1,
        main = "(b)")
lines(x, x, type = "l",lty=3)
```

R code to produce Figure 2.3.

```
par(mar = c(4, 5, 4, 4))
par(cex.axis=1.5, cex.main=1.5)
par(mfrow=c(1,2))
### SCAD WITH a=3.7 ###
y3 <- function(x){
  return(ifelse(abs(x)<=2*3, sign(x)*pmax((abs(x)-3),0),
             ifelse(abs(x)<=3.7*3 & abs(x) > 2*3, ((3.7-1)*x
             - sign(x)*3.7*3)/(3.7-2),
             ifelse(abs(x)>3.7*3, x,0 ))))}
plot(y3,xlim = c(-20,20), ylim = c(-20,20),
        xaxs="i", yaxs="i",
        ylab = expression(hat(beta)^s),
        xlab = expression(hat(beta)^o),
        font.main = 1,  main = "(a)")
lines(x, x, type = "l",lty=3)
abline(v = c(2*3,-3*2,3.7*3, -3*3.7), lty = 2)
### ADALASSO WITH LAMBDA = 3, GAMMA=2 ###
y5 = sign(x)*pmax((abs(x)-3/(abs(x)**(2))),0)
plot(x,y5,type='l', xlim= c(-10,10), ylim= c(-10,10),
        xaxs="i", yaxs="i",
        ylab = expression(hat(beta)^A),
        xlab = expression(hat(beta)^o),
        font.main = 1,  main = "(b)")
lines(x, x, type = "l",lty=3)
```

R Code for the simulation studies

R Code for producing results including plots for Example 1 up to Example 5 in Chapter 4 are accessible via the author's GitHub repositories. The corresponding files are:

- `ex1_code.R` for simulating Example 1.
- `ex2_code.R` for simulating Example 2.
- `ex3_code.R` for simulating Example 3.
- `ex4_code.R` for simulating Example 4.
- `ex5_code.R` for simulating Example 5.

R Code for data analysis on Diabetes data set

Similarly, for simplicity, the R code for performing the data analysis in Chapter 5 are accessible via the author's GitHub repositories under the file named `EDA_Diabetes.R`. The corresponding data used for Chapter 5 is from the file `Diabetes.csv`.

Bibliography

Breaux, H. J. (1967). On stepwise multiple linear regression, *Army Ballistic Research Lab Aberdeen Proving Ground MD* (10).
URL: *https://apps.dtic.mil/sti/pdfs/AD0658674.pdf*

Breiman, L. (1995). Better subset regression using the nonnegative garrote, *Technometrics* **37**(4): 373–384.
URL: *https://www.tandfonline.com/doi/abs/10.1080/00401706.1995.10484371*

Desboulets, L. (2018). A review on variable selection in regression analysis, *Econometrics* **6**(4): 45.
URL: *https://www.mdpi.com/2225-1146/6/4/45*

Donoho, D. L. and Elad, M. (n.d.). Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization.
URL: *https://www.pnas.org/doi/pdf/10.1073/pnas.0437847100*

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage, *Biometrika* **81**(3): 425–455.
URL: *https://academic.oup.com/biomet/article-abstract/81/3/425/256924*

Doornik, J. A. (2009). Econometric model selection with more variables than observations.
URL: *shorturl.at/bjzXY*

Draper, N. and Smith, H. (1966). *Applied regression analysis*, Wiley series in probability and mathematical statistics, Wiley, New York [u.a.].
URL: *https://onlinelibrary.wiley.com/doi/book/10.1002/9781118625590*

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression, *The Annals of Statistics* **32**(2): 407–499.
URL: *https://tibshirani.su.domains/ftp/lars.pdf*

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**(456): 1348–1360.
URL: *https://www.tandfonline.com/doi/abs/10.1198/016214501753382273*

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(5): 849–911.
URL: *https://pubmed.ncbi.nlm.nih.gov/19603084/*

Flom, P. L. and Cassell, D. L. (2007). Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use.
URL: *https://www.lexjansen.com/pnwsug/2008/DavidCassell-StoppingStepwise.pdf*

Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools, *Technometrics* **35**(2): 143.
URL: *https://www.jstor.org/stable/1269656*

Friedman, J., Hastie, T. and Tibshirani, R. (2010a). A note on the group lasso and a sparse group lasso, **22**.
URL: *https://arxiv.org/abs/1001.0736*

Friedman, J., Hastie, T. and Tibshirani, R. (2010b). Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software* **33**(1).
URL: *https://www.jstatsoft.org/article/view/v033i01*

Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso, *Journal of Computational and Graphical Statistics* **7**(3): 397.
URL: *https://www.jstor.org/stable/1390712*

Furnival, G. M. and Wilson, R. W. (1974). Regressions by leaps and bounds, *Technometrics* **16**(4): 499–511.
URL: *https://www.jstor.org/stable/1267601*

Hastie, Robert Tibshirani, J. F. (2008). *The Elements of Statistical Learning*, Springer, Stanford, California.
URL: *https://link.springer.com/book/10.1007/978-0-387-84858-7*

Hoerl, A. E. and Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **42**(1): 80–86.
URL: *https://www.jstor.org/stable/1271436*

Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples, *Biometrika* **76**(2): 297–307.
URL: *https://academic.oup.com/biomet/article/76/2/297/265326*

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2014). *An introduction to statistical learning: with applications in R*, Springer.
URL: *https://link.springer.com/book/10.1007/978-1-0716-1418-1*

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, pp. 1137–1143.
URL: *https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529*

Ogutu, J. O. and Piepho, H.-P. (2014). Regularized group regression methods for genomic prediction: Bridge, mcp, scad, group bridge, group lasso, sparse group lasso, group mcp and group scad, *BMC Proceedings* **8**(S5).
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4195413/*

Steyerberg, E. (1999). Stepwise selection in small data sets a simulation study of bias in logistic regression analysis, *Journal of Clinical Epidemiology* **52**(10): 935–942.
URL: *https://pubmed.ncbi.nlm.nih.gov/10513756/*

Thomas Lumley, A. M. (2020). *leaps: Regression Subset Selection.* R package version 3.1.
URL: *https://CRAN.R-project.org/package=leaps*

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1): 267–288.
URL: *https://tibshirani.su.domains/ftp/lasso-retro.pdf*

Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression, *The Annals of Applied Statistics* **2**(1).
URL: *https://projecteuclid.org/journals/annals-of-applied-statistics/volume-2/issue-1/Coordinate-descent-algorithms-for-lasso-penalized-regression/10.1214/07-AOAS147.full*

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1): 49–67.
URL: *https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2005.00532.x*

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty, *The Annals of Statistics* **38**(2).
URL: *https://projecteuclid.org/journals/annals-of-statistics/volume-38/issue-2/Nearly-unbiased-variable-selection-under-minimax-concave-penalty/10.1214/09-AOS729.full*

Zou, H. (2006). The adaptive lasso and its oracle properties, *Journal of the American Statistical Association* **101**(476): 1418–1429.
URL: *https://www.tandfonline.com/doi/abs/10.1198/016214506000000735*

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2): 301–320.
URL: *https://hastie.su.domains/Papers/B67.2%20(2005)%20301-320%20Zou%20%20Hastie.pdf*