Contents

Introduction **1**

1 Introduction to High-Dimensional Data **7**
1.1 Definition and Examples of High-Dimensional Data in bioscience 7
1.2 Challenge when Analysing High-dimensional Data 8
1.2.1 What goes wrong in high dimension setting ? 8
1.3 Statistical Suitable Methods for Analysing High-Dimensional Data 16
1.3.1 Ridge Regression 17
1.3.2 Lasso Regression 21
1.3.3 Dantzig Selector (DS) 29
1.3.4 Elastic-Net Regression 30
1.3.5 Selecting The Appropriate Tuning Parameter 31
1.4 Numerical Implementation 32
1.4.1 Simulated Data 32
1.4.2 Real data example 36

2 Measurement Error In Regression theory **40**
2.1 Introduction 40
2.1.1 Definition an motivating examples 40
2.1.2 Objective and some terminology 40
2.2 The Model Description 41
2.2.1 Estimated Coefficients and Behaviour of naive analyses 43
2.2.2 Correcting for Measurement Error in Multilinear regression 43

3 Measurement Error in High-Dimensional Context :Behaviour and Correction Methods **45**
3.1 Ridge Regression Estimation Over Measurement Error Ridden Data. 45
3.1.1 Ridge Regression Estimator of β and its Asymptotic Properties. 45
3.2 Measurement Error In Lasso 48
3.2.1 Impact Of Ignoring Measurement Error 49

3.2.2 Correction for Measurement Error in Lasso 50
3.2.3 Corrected Lasso (Non Convex Lasso) 51
3.2.4 Convex Conditional Lasso 52
3.2.5 Selecting The Tuning Parameter Under Measurement Error 55
3.3 Matrix uncertainty selector (MU-Selector) 56
3.4 Numerical Studies 60
3.4.1 Ridge under measurement error (simulation) 60
3.4.2 Measurement error with sparsity assumption (simulations): NCL, CoCo-Lasso and MUS implementation. 61

A R codes **62**
A.1 R code for numerical experiment of Lasso, Ridge and Elastic net. 62
A.2 R codes for real data example 69
A.3 R code for ridge regression over measurement error ridden data 72

List of Figures

1.1 An high-dimensional dataset. 8
1.2 Contour of the error and constraint function for the Ridge regression 19
1.3 Contour of the error and constraint function for the Lasso regression 22
1.4 plot showing how estimated coefficients for each methylated site change 38
1.5 Predicted Ages for each methods against the true Ages 38
1.6 *Cross-validation performance for Lasso.* 38
1.7 *Cross-validation performance for Ridge.* 38
1.8 Ridge versus Lasso coefficients paths 39
1.9 Coefficients paths elastic net 39
1.10 Cross-validation for elastic net regression 39
1.11 Lasso coefficients against elastic net coefficients 39

List of Tables

1.1 Two independent vectors 14
1.2 Two highly correlated vectors 14
1.3 Simulation results 34

3.1 Simulation results for ridge under measurement error 60

Introduction

This thesis is about measurement error in high-dimensional data.In recent decades, technological progress has led to a great abundance of data in many scientific fields.For example in genetics, a new framework has been developed, in which the number of variables **p** is larger than the number of observations **n** (high-dimensional data).High-dimensional data analysis has had a tremendous growth in popularity and a plethora of methods has been proposed for statistical modelling of, and inference in high-dimensional data.Penalized regression methods such as ridge regression [17], Lasso [21] methods and Dantzig selector [4] are particularly good in this context.

In almost all disciplines, it may not be possible to observe a variable accurately, for some reason, and therefore it is necessary to work with an error-prone version of that variable.Any measurement process can be affected by errors, usually due to the measuring instrument or the sampling process.The consequences of ignoring measurement error, many of which have been known for some time, can range from the non-existent to the rather dramatic.Throughout this work, attention is given to the effects of measurement error on analyses that ignore it. This is mainly because the majority of researchers do not account for measurement error, even if they are aware on its presence and potential impact.In part this is because the information or extra data needed to correct for measurement error may not be available.Typically , when measurement error creep into the data, there are tree main reason why measurement error cannot be ignored; it can cause bias in parameter estimation [3], interfere with variable selection [20] and lead to a loss of power [5] leading to trouble in detecting relationships among variables.Results on the bias of naive estimators often provide the added bonus of suggesting a correction method.

Applying high-dimensional regression methods that do not correct for measurement errors result in faulty inference as demonstrated for the Lasso [19]. Consequently, correction for measurement error in penalized regression has recently been studied by various authors.Example include ; "Ridge regression approach to measurement error"[19] , Non Convex Lasso (NCL) by Loh and Wainwright [20], the Convex Conditional Lasso (CoCoLasso) of Datta and Zou [8] and the Matrix Uncertainty Selector proposed by Rosenbaum and Tsybakov (MUS) [18].

The organization of this thesis is as follows; **Chapter 1** presents high-dimensional data together with potential challenges when analysing the later, along with some statistical methods

one may use to handle this kind of datasets. **Chapter 2** introduces the measurement error in regression theory, provides an overview of the consequences of measurement error in linear regression and introduces some corrections methods. **Chapter 3** describes behaviour of measurement error in high-dimensional regression and introduces some high-dimensional approaches (methods) to correct for measurement error in high-dimensional context. Both real and simulated data are used for illustrations.

Chapter 1

Introduction to High-Dimensional Data

1.1 Definition and Examples of High-Dimensional Data in bio-science

High-dimensional data are defined as data in which the number of features (*variables observed*) **p**, are close to or large than the number of observations (or *data points*) **n**. The opposite is **low-dimensional data**, in which the number of observations **n**, far outnumbers the number of feature **p**.

A related concept is **Wide data** which refers to data with numerous features irrespective of the number of observations; similarly, **tall data** is often used to denote data with large number of observations. This concept should not be therefore confuse with notion of **big data** which is data that contains greater *variety*, arriving in increasing *volumes* and with more *velocity* known as the threes **Vs** (visit, `https://www.oracle.com/big-data/what-is-big-data/`).

High-dimensional datasets are become more common in many scientific fields as new automated data collection techniques have been developed.And example in biological sciences may include *data collected from hospital patients recording symptoms, blood test results, behaviours and general health* resulting in datasets with large number of features.
And example of what high-dimensional data might look like in a biomedical study is shown in figure 1.1 below. Here are examples of descriptions of research questions whose associate datasets can be considered as high-dimensional data:

- predicting patient blood pressure using: *cholesterol level in blood,age and BMI as well as information on 200000 single nucleotide polymorphisms from 100 patients*
- Predicting probability of a patient's cancer progressing using: *gene expression data from 20000 genes as well as data associated with general patient health (age, weight,BMI, blood pressure) and cancer growth (tumour , localised spread,blood test results)*

Example of application, including in social science are extremely numerous; see **Plomin (2018)**.

	Blood pressure	Heart rate	Respiratory rate	Platelets	Lymphocites	Red cells	BMI	survival	age	Body fat	cholesterol	.... + 20000 genes expression
Patient 1	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 2	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 3	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 4	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 5	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 6	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 7	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 8	....	....	....	....	....	....	....	....	....	....	....	... ...
Patient 9	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...
....	....	....	....	....	....	....	....	....	....	....	....	... ...

Figure 1.1: an overview of an high-dimensional dataset with P=20011 features and n=200 observations

1.2 Challenge when Analysing High-dimensional Data

Analyses of high-dimensional data require consideration of potential problems that come with having more features than observations.Such datasets pose a challenge for data analysis as standard methods of analysis, such as *least squares linear regression*, are no longer appropriate.Many of the issues that arise in the analysis of high-dimensional data are know in classical approaches, since they apply also when $n > p$: these include the role *bias-variance trade-off* and the danger of *over-fitting*.Though these issues are always relevant, they can become particularly important when the number of features is very large relative to the number of observations.

1.2.1 What goes wrong in high dimension setting ?

In other to illustrate the need for extra care and specialized technique for regression when $p > n$, we begin by examining what can go wrong if we apply a statistical technique not intended for high-dimensional setting. For this purpose, we examine *least squares regression*.But the same concepts apply to *logistic regression, linear discriminant analysis* and other classical statistical approaches.

Setup of Linear Regression Model

The general form of the multiple linear regression model is as follows:

$$Y = \mathbb{E}[Y|X] + \epsilon \;=\; \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon \tag{1.1}$$

Where y is the dependent variable, $\beta_0, \beta_1, ..., \beta_p$ are regressions coefficients, and $X_1, ..., X_p$ are independents variables in the model; $\mathbb{E}[Y]$ the expectation of the response variable. In the classical regression setting, it is usually assumed that the error term ϵ follows the *normal distribution* with mean $\mathbb{E}[\epsilon] = 0$ and constant variance $Var[\epsilon] = \sigma^2$.

We consider a datasets from the following model

$$Y_i = \beta_0 + \beta_1 X_{i1} + ... + \beta_p X_{ip} + \epsilon_i,\ i = 1, ..., n \tag{1.2}$$

Where X_{ij} is the j^{th} variable for individual i and $\epsilon_i's$ are random errors assuming $\mathbb{E}[\epsilon_i] = 0$ and $Var[\epsilon_i|X] = \sigma^2\ for\ i = 1, 2, ..., n$. The data from this model can be written in matrix form:

$$y = X\beta + \epsilon, \tag{1.3}$$

where:

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},\ X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix},\ \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix},\ and\ \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

The regression parameter are estimated by minimizing ordinary least squares:

$$\sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 X_{i1} + ... + \beta_p X_{ip})]^2 = (y - X\beta)^t (y - X\beta) = \| y - X\beta \|^2,\ (\| . \|\ ^1).$$

Ordinary Least Squares Estimates (OLS Estimates)

Proposition 1.2.1 (*from [23]*). *The least squares estimation of β for linear regression model is given by,*

$$b = argmin_\beta \left\{ \| y - X\beta \|_2^2 \right\} = (X^t X)^{-1} X^t y, \tag{1.4}$$

assuming $(X^t X)$ is a non-singular matrix.Note that this is equivalent to assuming that the matrix X is of full rank[2].

proof. we need to minimize the residual sum of squares by solving the following equation: $\frac{\partial}{\partial \beta}(\| y - X\beta \|^2) = 0$; set $f(\beta) = \| y - X\beta \|^2$,

$$\begin{aligned} f(\beta) &= \| y - X\beta \|^2 \\ &= \| y \|^2 + \| X\beta \|^2 - 2 < y, X\beta > = y^t y + \beta^t X^t X\beta - 2y^t X\beta = y^t y + \beta^t X^t X\beta - 2(X^t y)^t \beta \end{aligned}$$

[1] $\| . \|$ is the Euclidian norm on $\mathbb{R}^n$

[2] i.e, $rank(X) = p + 1 < n$, this then implies that $rank(X^t X) = p + 1$ and therefore that $X^t X$ is invertible.

(by setting $X^t y = (w_j)_{1\leq j\leq} \in \mathcal{M}_{n\times 1}$, $X^t X = (a_{ij})_{1\leq i,j\leq p+1} \in \mathcal{M}_{p+1}$ and recalling that $\beta^t X^t X\beta$ is a quadratic form[3])

$$= \sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{p+1} a_{ii}\beta_i^2 + 2\sum_{1\leq i<j\leq p+1} \beta_i a_{ij}\beta_j - 2\sum_{i=1}^{n} w_i y_i$$

by taking partial derivative with respect to each component of β, we obtain $\frac{\partial}{\partial b} f(b) = -2X^t y + 2X^t Xb$,

$$\frac{\partial}{\partial b} f(b) = 0 \Rightarrow b = (X^t X)^{-1} X^t y \quad \textit{as required.}$$

Proposition 1.2.2. *The estimator $b = (X^t X)^{-1} X^t y$ is and unbiased estimator of β. In addition,its covariance matrix is given by*

$$Cov(b) = (X^t X)^{-1}\sigma^2.$$

proof.

$$\mathbb{E}[b] = \mathbb{E}[(X^t X)^{-1} X^t y] = (X^t X)^{-1} X^t \mathbb{E}[y] = (X^t X)^{-1} X^t (X\beta) = \beta.$$

this complete the proof of unbiasness of b.

$$\begin{aligned} Cov(b) &= Cov[(X^t X)^{-1} X^t y] = [(X^t X)^{-1} X^t] Cov(y) [(X^t X)^{-1} X^t]^t \\ &= [(X^t X)^{-1} X^t]\sigma^2 \mathbb{I}_n [(X^t X)^{-1} X^t]^t = (X^t X)^{-1}\sigma^2. \textit{ as required.} \end{aligned}$$

In order to estimate σ^2, we consider the residual sum of square (RSS)

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \| \, y - \hat{y} \, \|^2$$

using (1.4)

$$\begin{aligned} &= (y - Xb)^t (y - Xb) = [y - X(X^t X)^{-1} X^t y]^t [y - X(X^t X)^{-1} X^t y] \\ &= y^t [\mathbb{I}_n - X(X^t X)^{-1} X^t] y = y^t P y \end{aligned}$$

which is actually the distance measure between observe y and fitted regression value $\hat{y} = Xb$.

$P = [1 - X(X^t X)^{-1} X^t]$ is and idempotent matrix[4].

Proposition 1.2.3 (*from [23]*)**.** *The unbiased estimator of the variance σ^2 in the multiple linear regression is given by:*

$$s^2 = \frac{RSS}{n-p-1} = \frac{1}{n-p-1}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{1.5}$$

[4] i.e ,$p^2 = p$

Before proving this assertion, let's recall the following lemmas:

Lemma 1. *Let $A_{n\times n}$ be and idempotent matrix of rank p then the eigenvalues of A are either* 1 *or* 0.

proof. Let λ_i and v_i be the eigenvalue and the corresponding normalized eigenvector ($\parallel v_i \parallel = 1$) of matrix A, respectively.

We then have $Av_i = \lambda_i v_i$, and $v_i^t A v_i = v_i^t(\lambda_i v_i) = \lambda_i \parallel v_i \parallel^2 = \lambda_i$. on the other hand, since $A^2 = A$, $v_i^t A v_i = v_i^t A^2 v_i = (A^t v_i)^t (A v_i) = (\lambda_i v_i)^t(\lambda_i v_i) = \lambda_i^2 v_i^t v_i = \lambda_i^2 \parallel v_i \parallel^2 = \lambda_i^2$; (*recall that A and A^t* have sames eigenvalues).This end the proof.

Lemma 2. *If A is and idempotent matrix, then $tr(A) = rank(A) = p$.*

proof. If the rank of and $n \times n$ matrix is p, then A has p eigenvalues of 1 and $n - p$ eigenvalues of 0, since the eigenvalues of A are either 1 or 0.Thus we can write $rank(A) = \sum_{i=1}^{n} \lambda_i = p$. from matrix theory, there is an orthogonal matrix V such that

$$V^t A V = \begin{pmatrix} \mathbb{I}_p & 0 \\ 0 & 0 \end{pmatrix}$$

therefore, we have $tr(A) = tr(V^t V A) = tr(V^t A V) = p = rank(A)$. (*here we use the simple fact that tr(AB)=tr(BA) for any matrix A,B*)

Lemma 3. *Let $y^t = (y_1, y_2, ..., y_n)$ be an $n \times 1$ vector with mean $\mu^t = (\mu_1, ..., \mu_n)$ and variance σ^2 for each component. Further, it is assumed that $y_1, y_2, ..., y_n$ are independent.Let A be and $n \times n$ matrix.*

The expectation of the quadratic form of random variables is given by:

$$\mathbb{E}[y^t A y] = \sigma^2 tr(A) + \mu^t A \mu, \tag{1.6}$$

proof. first we observe that,

$(y-\mu)^t A(y-\mu) = y^t A y - y^t A \mu - \mu^t A y + \mu^t A \mu = y^t A y - 2\mu^t A y + \mu^t A \mu = y^t A y - 2\mu^t A(y - \mu) - \mu^t A \mu$. Thus , $y^t A y = (y-\mu)^t A(y-\mu) + 2\mu^t A(y-\mu) + \mu^t A \mu$. We write:

$$\begin{aligned}
\mathbb{E}[y^t A y] &= \mathbb{E}[(y-\mu)^t A(y-\mu)] + 2\mathbb{E}[\mu^t A(y-\mu)] + \mu^t A \mu \\
&= \mathbb{E}[\sum_{i,j}^{n} a_{ij}(y_i - \mu_i)(y_j - \mu_j)] + 2\mathbb{E}[\sum_{i,j}^{n} \mu_i a_{ij}(y_j - \mu_j)] + \mu^t A \mu \\
&= \mathbb{E}[\sum_{i}^{n} a_{ii}(y_i - \mu_i)^2] + \mathbb{E}[\sum_{i \neq j} a_{ij}(y_i - \mu_i)(y_j - \mu_j)] + 2\mathbb{E}[\sum_{i,j}^{n} \mu_i a_{ij}(y_j - \mu_j)] + \mu^t A \mu
\end{aligned}$$

using the fact that $y_1, y_2, ..., y_n$ are independent,

$$= \sum_i^n a_{ii} \underbrace{\mathbb{E}[(y_i - \mu_i)^2]}_{\sigma^2} + \sum_{i \neq j} a_{ij} \underbrace{\mathbb{E}[(y_i - \mu_i)]\mathbb{E}[(y_j - \mu_j)]}_{0} + 2\sum_{i,j}^n \mu_i a_{ij} \underbrace{\mathbb{E}[(y_j - \mu_j)]}_{0} + \mu^t A \mu$$

$$= \sigma^2 \sum_i^n a_{ii} + \mu^t A \mu = \sigma^2 tr(A) + \mu^t A \mu. \; asrequired.$$

proof. (*of the Proposition 1.2.2*)

$\mathbb{E}[s^2] = \frac{1}{n-p-1}\mathbb{E}[RSS] = \frac{1}{n-p-1}\mathbb{E}[y^t P y] = \frac{1}{n-p-1}(\sigma^2 tr(P) + (\mathbb{E}[y])^t P(\mathbb{E}[y]))$ (by *lemma(3)*). Since $P = [\mathbb{I}_n - X(X^tX)^{-1}X^t]$ and $X(X^tX)^{-1}X^t$ are idempotent matix, using lemma(2) ,we have $rank(X(X^tX)^{-1}X^t) = tr(X(X^tX)^{-1}X^t) = tr(X^{tX}(X^tX)^{-1}) = tr(\mathbb{I}_{p+1}) = p+1$. since $tr(A+B) = tr(A) + tr(B)$, we have $tr(P) = tr(\mathbb{I}_n) - tr(X(X^tX)^{-1}X^t) = n-p-1$. Recalling that $(\mathbb{E}[y]) = X\beta$, we finally obtain

$$\begin{aligned}\mathbb{E}[s^2] &= \frac{1}{n-p-1}[\sigma^2(n-p-1) - (X\beta)^t[\mathbb{I}_n - X(X^tX)^{-1}X^t](X\beta)] \\ &= \frac{1}{n-p-1}[\sigma^2(n-p-1) - (X\beta)^t[(X\beta) - (X\beta)]] = \sigma^2\end{aligned}$$

and we have the result.

Assessing The Accuracy of The Model (*visit [14]*)

Once the parameters of the model have been estimated, It is natural to want to quantify the extent to which the model fits the data.The quality of a linear regression fit is typically assessed using two related quantities: the "*Residual Standard Error" (RSE)* and the R^2 statistic.

Residual Standard Error (RSE): Recall from the model (1.2) that associated with each observation an error term ϵ .Due to the presence of these error terms, even if we knew the true regression line (i.e even if $\beta^t = (\beta_0, ..., \beta_p)$ where known) , we would not be able to perfectly predict Y from X.

The RSE is an estimate of the standard deviation of ϵ.Roughly speaking, it is the average amount that the response will deviate from the true regression line.It is computed using the formula

$$RSE = \sqrt{s^2} = \sqrt{\frac{1}{n-p-1}\sum_{i=1}^n (y_i - \hat{y}_i)^2}. \tag{1.7}$$

The RSE is considered a measure of the lack of fit of the model (1.2) to the data. If the model are very close to the outcomes values, that is, if $\hat{y}_i \simeq y_i$ for $i = 1, ..., n$, then (1.7) will be small, and we can conclude that the model fits the data very well.On the other hand, if $\hat{y}_i$ is very far from y_i for one or more observations, then the RSE may be quite

large, indicating that the model doesn't fit the data very well.

R^2 Statistic: The RSE provides an absolute measure of lack of fit of the model to the data.But since it is measured in the unit of Y, it is not always clear what constitutes a good RSE. The R^2 Statistic provides an alternative measure of fit.It take a form of a proportion: The proportion of variance explained and so it always takes on a value between 0 and 1; an is independent of the scale Y.
To calculate R^2 , we use the formula

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} \tag{1.8}$$

where, $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ is the total sum of squares, and RSS is defined in (1.5).
TSS measures the total variance in the response Y, and can be thought as the amount of variability inherent in response before the regression is performed. In contrast, RSS measures the amount of variability that is left unexplained after performing the regression. Hence,$TSS - RSS$ measures the amount of variability in the response that is explained by performing the regression; and R^2 measure the proportion of variability in Y that can be explained using X.

An R^2 Statistic that is close to 1 indicates that a large proportion of the variability in the response is explained by the regression. A number near 0 indicate that the regression does not explain much of the variability in the response; this might occur because the linear model is wrong or the error variance σ^2 is high.

Now that a brief presentation of the linear model has been made, come back to the main question to know the problems encountered in high-dimension setting.

Theoretically: When $p > n$, X^tX is not invertible (or near singular) and s^2 in (1.5) is not defined.

i) visualisation problem: Datasets with large number of features are difficult to visualise.When exploring low-dimensional datasets, it is possible to plot the response variable against each of the limited number of explanatory variables to get an idea which of these are important predictors of the response.With high-dimensional data,the large number of explanatory variables makes doing this difficult.

ii) Risk of Overfitting: When the number of features p is as large as or large than, the number of observations ($p \geq n$), least squares as described previously should not be performed.The reason is simple; regardless of whether or note there truly is a relationship between the features and the response, least square will yield a set of coefficient estimates that result in a perfect fit to the data, such that the residuals are zero. In other

words, though it is possible to perfectly fit the training data in high-dimensional setting, the resulting linear model will perform extremely poorly on an independent text set and therefore does not constitute a useful model.

This indicates the importance of applying extra care methods when analysing data sets with a large number of variables, and of always evaluating model performance on an independent test set.

iii) Multi-Collinearity problem: Another problem in carrying out regression on high-dimensional data is dealing with multi-collinearity.

Definition 1.2.1. *In multilinear regression, collinearity refers to the situation in which two or more variables are highly correlated.*

The large numbers of features in these datasets makes high correlations between variables more likely. Consider the regression model (1.1) , the collinearity occurs if the independent variable X_i is highly linearly correlated to another one or more independent variable $(X_j)_{1\leq j\neq i\leq p}$; in other words,the i^{th} column of X can be *almost* linearly express by one or more other column vectors in X.

If there is a perfect collinearity among column vectors of X then the matrix X^tX is not invertible.Therefore , it is problematic to solve for the unique least squares estimators of the regression coefficients from the normal equation (1.4) $b = (X^tX)^{-1}X^ty$.

when the column vectors of the design matrix X are highly correlated, then the matrix X^tX becomes ill-conditioned[5] or near singular and the least squares estimator become less reliable even though we can find a unique solution of the normal equation.To see this let's look at the following example of two simple data sets:

x_1	x_2
10	10
10	10
10	15
10	15
15	10
15	10
15	15
15	15

Table 1.1: Two independent vectors

x_1	x_2
10	10
11	11.4
11.9	12.2
12.7	12.5
13.3	13.2
14.2	13.9
14.7	14.4
15	15

Table 1.2: Two highly correlated vectors

[5]its condition number is too large, see " `https://arxiv.org` " for more detail

The correlation matrix of vectors in the first example datasets table 1.1 is 2×2 identity matrix

$$X^tX = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (X^tX)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The correlation matrix of the two vectors in the second example datasets table 1.2 is ,

$$X^tX = \begin{pmatrix} 1 & 0.99215 \\ 0.99215 & 1.0000 \end{pmatrix}, \quad (X^tX)^{-1} = \begin{pmatrix} 63.94 & -64.44 \\ -64.44 & 63.94 \end{pmatrix}$$

recall that for linear regression, $Cov(b) = (X^tX)^{-1}\sigma^2$ (see proposition 1.2.1)) ; for the vector in the second example dataset we have

$$Cov(b) = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}, \; b = (b_1, b_2)^t \quad say, \; Var(b_1) = Var(b_2) = \sigma^2,$$

for the example in the second example dataset, we have

$$Var(b_1) = Var(b_2) = 63.94 \times \sigma^2,$$

The variances of the regression coefficients are *inflated* in the second datasets; this is because of the collinearity of the two vectors in the dataset.

Lemma 4. *An $n \times n$ ill-conditioned or near singular matrix has at least one of its eigenvalues close to zero, and then the eigenvalue of the inverse tend to be very large.*

Proposition 1.2.4 (*from [23]*)**.** *The average Euclidean distance measure $\mathbb{E}[\| b - \beta \|^2]$ between the least squares estimate b and the true parameter β is given by:*

$$\mathbb{E}[\| b - \beta \|^2] = \sigma^2 tr[(X^tX)^{-1}] \tag{1.9}$$

proof.

$$\mathbb{E}[\| b-\beta \|^2] = \mathbb{E}[\| b \|^2 + \| \beta \|^2 - 2 < b, \beta >] = \mathbb{E}[\| b \|^2] + \| \beta \|^2 - 2\beta^t \underbrace{\mathbb{E}[b]}_{\beta} = \mathbb{E}[\| b \|^2] - \| \beta \|^2,$$

$$\begin{aligned} \mathbb{E}[\| b \|^2] &= \mathbb{E}[b^t b] = \mathbb{E}[(X^tX)^{-1}X^ty)^t(X^tX)^{-1}X^ty] \\ &= \mathbb{E}[y^t(X(X^tX)^{-1}(X^tX)^{-1}X^t)y] \end{aligned}$$

using lemma(3), we get

$$
\begin{aligned}
&= \sigma^2 tr[X(X^tX)^{-1}(X^tX)^{-1}X^t] + \mathbb{E}[y]^t(X(X^tX)^{-1}(X^tX)^{-1}X^t)\mathbb{E}[y] \\
&= \sigma^2 tr[X(X^tX)^{-1}(X^tX)^{-1}X^t] + (X\beta)^t(X(X^tX)^{-1}(X^tX)^{-1}X^t)(X\beta) \\
&= \sigma^2 tr[(X^tX)^{-1}] + \beta^t\beta = \sigma^2 tr[(X^tX)^{-1}] + \parallel \beta \parallel^2
\end{aligned}
$$

Remark 1.2.1. *Assuming that* (X^tX) *has k distinct eigenvalues* $\lambda_1, ..., \lambda_k$ *, then the eigenvalues of* $(X^tX)^{-1}$ *are* $\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_k}$*, denoting by* $V = (v_1, ..., v_k)^t$ *the corresponding normalized eigenvectors, we can write* $V^t(X^tX)^{-1}V = D = diag(\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_k})$.
Moreover, $tr(X^tX)^{-1}) = tr(V^tV(X^tX)^{-1}) = tr(V^t(X^tX)^{-1}V) = tr(D) = \sum_{i=1}^{k} \frac{1}{\lambda_i}$ *; we then have:*

$$\mathbb{E}[\parallel b - \beta \parallel^2] = \sigma^2 \sum_{i=1}^{k} \frac{1}{\lambda_i} \Leftrightarrow \mathbb{E}[\parallel b \parallel^2] = \parallel \beta \parallel^2 + \sigma^2 \sum_{i=1}^{k} \frac{1}{\lambda_i}. \tag{1.10}$$

Now it is easy to see that if one of λ_i, $i = 1, ..., k$ is very small, say for instance $\lambda_i = 0.00001$ then roughly, $\parallel b \parallel^2 = \sum_{i=1}^{k} b_i^2$ may **over estimate** $\parallel \beta \parallel^2 = \sum_{i=1}^{k} \beta_i^2$ by $10000\sigma^2$ times.
The above discussions indicate that if some columns in X are highly correlated with other column in X then, from *lemma(4)* , the covariance matrix $Cov(b) = (X^tX)^{-1}\sigma^2$ will have one or more large eigenvalues so that the mean Euclidean distance of $\mathbb{E}[\parallel b - \beta \parallel^2]$ will be inflated.Consequently, this makes the estimation of regression parameter β less reliable.Thus the high levels correlation between variable in high-dimensional datasets will have negative impact on least square estimates of regression parameter.

Clearly, alternative approaches that are better-suited to the high-dimensional setting are required.

1.3 Statistical Suitable Methods for Analysing High-Dimensional Data

As we found out in the above challenges, carrying out linear regression on datasets with large numbers of features is difficult due to:

High levels of correlation between variables ,difficulty to identifying a clear response by visualizing and risk of over-fitting.This problems are common to the analysis of many high-dimensional datasets, for example, those using genomics data with multiples genes. While straightforward linear regression cannot be used in datasets with many features, high-dimensional regression methods are available with methods to deal with multicollinearity , over-fitting and fitting models including many explanatory variables.

1.3.1 Ridge Regression

Ridge regression is one of the remedial measures for handling severe multicollinearity in least squares estimation.Multicollinearity occurs when the predictors included in the linear model are highly correlate with each other.When this is the case, the matrix X^tX tends to be singular or ill-conditioned and hence identifying the least squares estimates will encounter numerical problems.

To motivate the Ridge estimator, we first take a look at the **Mean Square Error**[6] (MSE) ,$MSE(b) = \mathbb{E}[\| b - \beta \|^2]$ of least squares estimator of β, which can be break into two parts: $bias^2 + variance$.

Proposition 1.3.1. $\mathbb{E}[\| b - \beta \|^2] = \sum_{j=1}^{n}(\mathbb{E}[b_j] - \beta_j)^2 + \sum_{j=1}^{n} Var[b_j]$

proof.

$$\begin{aligned}\mathbb{E}[\| b - \beta \|^2] &= \sum_{j=1}^{n} \mathbb{E}[(b_j - \beta_j)^2] = \sum_{j=1}^{n} \mathbb{E}[b_j^2] + \beta_j^2 - 2\beta_j \mathbb{E}[b_j] \\ &= \sum_{j=1}^{n} Var[b_j] + \mathbb{E}[b_j]^2 + \beta_j^2 - 2\beta_j \mathbb{E}[b_j] = \sum_{j=1}^{n}(\mathbb{E}[b_j - \beta_j)^2 + \sum_{j=1}^{n} Var[b_j]\end{aligned}$$

According to "Gauss-Markov" theorem, the least squares approach achieves the smallest variance among all unbiased linear estimates.This however does not necessarily guarantee the minimum **MSE**.

To better distinguish different type of estimators, let $\hat{\beta}^{LS}$ denote the ordinary least square estimator of β. We shown that $MSE(\hat{\beta}^{LS}) = \mathbb{E}[\| \hat{\beta}^{LS} - \beta \|^2] = \sigma^2 tr[(X^tX)^{-1}]$ (1.9) thus,$\mathbb{E}[\| \hat{\beta}^{LS} \|^2] = \| \beta \|^2 + \sigma^2 tr[(X^tX)^{-1}$ (1.10); it can be seen that, with ill-conditioned X^tX, the resultant LSE $\hat{\beta}^{LS}$ would be large in length $\| \hat{\beta}^{LS} \|$ and associated with inflated standard error (see (1.10)). This inflated variation would lead to poor model prediction as well.

The Ridge regression is a constrained version of least squares.It tackles the estimation problem by providing biased estimator yet with small variance.

Ridge Shrinkage Estimator

Theorem 1.3.1. *For any estimator b, the least squares criterion* $\mathcal{Q}(b) = \| y - Xb \|^2$ *can be rewritten as its minimum, reached at* $\hat{\beta}^{LS}$ *plus a quadratic form in b.*

[6] MSE is a commonly-used measured for assessing quality of estimation

proof.

$$\begin{aligned}
\mathcal{Q}(b) &= \| y - Xb \|^2 = \| y - X\hat{\beta}^{LS} + X\hat{\beta}^{LS} - Xb \|^2 \\
&= \| y - X\hat{\beta}^{LS} \|^2 + \| X\hat{\beta}^{LS} - Xb \|^2 + 2 < X\hat{\beta}^{LS} - Xb, y - X\hat{\beta}^{LS} > \\
&= \mathcal{Q}_{min} + (X\hat{\beta}^{LS} - Xb)^t (X\hat{\beta}^{LS} - Xb) + 2(< X\hat{\beta}^{LS} - Xb)^t (y - X\hat{\beta}^{LS}) \\
&= \mathcal{Q}_{min} + \underbrace{(\hat{\beta}^{LS} - b)^t X^t X (\hat{\beta}^{LS} - b)}_{\phi(b)} + \underbrace{2(\hat{\beta}^{LS} - b)^t X^t (y - X\hat{\beta}^{LS})}_{A}
\end{aligned}$$

$$\begin{aligned}
A &= 2(\hat{\beta}^{LS} - b)^t X^t (y - X(XX^t)^{-1} X^t y) = 2(\hat{\beta}^{LS} - b)^t [X^t y - (XX^t)((XX^t)^{-1}) X^t y] \\
&= 2(\hat{\beta}^{LS} - b)^t [X^t y - X^t y] = 0
\end{aligned}$$

$$thus, \quad \mathcal{Q}(b) = \mathcal{Q}_{min} + \phi(b) \tag{1.11}$$

contour for each constant of the quadratic form $\phi(b)$ are hyper-ellipsoids centred at ordinary LSE $\hat{\beta}^{LS}$. In view of (1.10), it is reasonable to expect that, if one move away from $\mathcal{Q}_{min}$, the movement is in a direction which shortens the length of b.

The optimization problem in Ridge regression can be state as:

$$minimize \| \beta \|^2 \ \ subject\ to\ (\hat{\beta}^{LS} - b)^t X^t X (\hat{\beta}^{LS} - b) = \phi_0\ for\ some\ constant\ \phi_0.$$

The enforced constrain guarantees a relatively small residual sum of squares $\mathcal{Q}(\beta)$ when compared to its minimum $\mathcal{Q}_{min}$.As a Lagrangian problem, it is equivalent to

$$minimizing\ f(\beta) = \| \beta \|^2 + \frac{1}{k}[(\hat{\beta}^{LS} - b)^t X^t X (\hat{\beta}^{LS} - b) - \phi_0], \qquad k > 0$$

Where $\frac{1}{k}$ is the multiplier chosen to satisfy the constraint.

Proposition 1.3.2 (Hoerl and Kennard (1970))**.** *The numerical solution of this problem corresponding to the Ridge regression estimator of β is,*

$$\hat{\beta}^R = (X^t X + k\mathbb{I}_p)^{-1} X^t y \tag{1.12}$$

proof. $f(\beta) = \sum_i \beta_i^2 + \frac{1}{k}\left[\sum_i (\beta_i - \hat{\beta}_i^{LS})^2 a_{ii} + 2\sum_{1 \le i < j \le n} (\beta_i - \hat{\beta}_i^{LS}) a_{ij} (\beta_j - \hat{\beta}_j^{LS}) - \phi_0 \right]$

by taking partial derivative with respect to each component of β, we obtain $\frac{\partial f(\beta)}{\partial \beta} = 2\beta + \frac{1}{k}(2X^t X(\beta - \hat{\beta}^{LS}))$,

$$\begin{aligned}
&\frac{\partial f(\beta)}{\partial \beta} = 0 \Rightarrow 2\beta + \frac{1}{k}(2X^t X\beta) = \frac{2}{k} X^t X \hat{\beta}^{LS} \\
&\qquad \Rightarrow \beta = (X^t X + k\mathbb{I}_p)^{-1} X^t X \hat{\beta}^{LS} = (X^t X + k\mathbb{I}_p)^{-1} X^t X (X^t X)^{-1} X^t y = (X^t X + k\mathbb{I}_p)^{-1} X^t y
\end{aligned}$$

An equivalent way is to write the Ridge problem in the penalized or constrained least squares form by :

$$minimize \ \| y - X\beta \|^2 \quad subject\ to \ \| \beta \|^2 \leq s \ for\ some\ constant\ s \tag{1.13}$$

the Lagrangian problem become

$$minimizing \ \| y - X\beta \|^2 + \lambda \| \beta \|^2 \tag{1.14}$$

which yield the same estimator given in (1.12).The penality parameter $\lambda \geq 0$ controls the amount of shrinkage in $\| \beta \|^2$.The large value of λ, the greater amount of shrinkage.For this reason, the Ridge estimator is also called the shrinkage estimator. There is one-to-one correspondence among λ, s, k *and* ϕ_0.

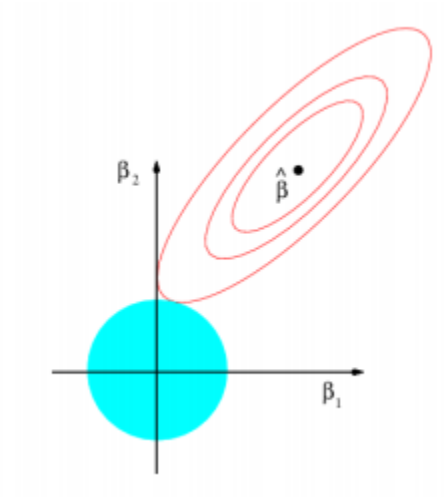

Figure 1.2: contour of the error and constraint function for the Ridge regression in two dimensional case. The solid blue area is the constraint region $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipse is the contour of RSS. **(** *figure from*. [**14**], *P*.244)

Why does Ridge regression improve over Least Square ?

We shall compute the expectation and variance of $\hat{\beta}^R$.Eventually, we want to compare $\hat{\beta}^R$ with $\hat{\beta}^{LS}$ to see whether a smaller MSE can be achieve by $\hat{\beta}^R$ for certain values of k.

Remark 1.3.1. $\hat{\beta}^R = (X^tX + k\mathbb{I}_p)^{-1}X^ty = (\mathbb{I}_p + k(X^tX)^{-1})^{-1}(X^tX)^{-1}X^ty = (\mathbb{I}_p + k(X^tX)^{-1})^{-1}\hat{\beta}^{LS}$. *denote* $Z = (\mathbb{I}_p + k(X^tX)^{-1})^{-1}$.

$$\mathbb{E}[\hat{\beta}^R] = \mathbb{E}[Z\hat{\beta}^{LS}] = Z\mathbb{E}[\hat{\beta}^{LS}] = Z\beta, \quad \hat{\beta}^R \ is\ a\ biased\ estimator \tag{1.15}$$

$$Cov(\hat{\beta}^R) = ZCov(\hat{\beta}^{LS})Z^t = Z(X^tX)^{-1}\sigma^2 Z^t = \sigma^2[Z(X^tX)^{-1}Z^t] \tag{1.16}$$

Let $\lambda_{max} = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p = \lambda_{min}$ denote the eigenvalues of X^tX, then the corresponding eigenvalues of Z are $\frac{\lambda_j}{\lambda_j+k}$, $j = 1, ..., p$. From (1.10), $MSE(\hat{\beta}^{LS}) = \sigma^2 \sum_{j=1} \frac{1}{\lambda_j}$.

Proposition 1.3.3 (*from [?]*). *If $MSE(\hat{\beta}^R, k)$ denote the mean square error of ridge regression estimator, then*

$$MSE(\hat{\beta}^R, k) = k^2\beta^t(X^tX + k\mathbb{I})^{-2}\beta + \sigma^2 \sum_j \frac{1}{\lambda_j}.\frac{\lambda_j^2}{(\lambda_j+k)^2} = \lambda_1(k) + \lambda_2(k). \tag{1.17}$$

proof. For the component in Ridge estimator, it can be found from (1.15) and (1.16) that the sum of their square biases is:

$$\begin{aligned}\sum_j (\mathbb{E}[\hat{\beta}_j^R] - \beta_j)^2 &= (\mathbb{E}[\hat{\beta}^R] - \beta)^t(\mathbb{E}[\hat{\beta}^R] - \beta) = (Z\beta - \beta)^t((Z\beta - \beta) \\ &= \beta^t(\mathbb{I} - Z)^t(\mathbb{I} - Z)\beta\end{aligned}$$

we have $Z^t = Z$, thus

$$\begin{aligned}&(\mathbb{I} - Z)^t = \mathbb{I} - Z = \mathbb{I} - (\mathbb{I} + k(X^tX)^{-1})^{-1} \\ &= [\mathbb{I} + k(X^tX)^{-1}]^{-1}[\mathbb{I} + k(X^tX)^{-1}] - [\mathbb{I} + k(X^tX)^{-1}] \\ &= [\mathbb{I} + k(X^tX)^{-1}](\mathbb{I} + k(X^tX)^{-1} - \mathbb{I}) = [(X^tX)^{-1}(X^tX + k\mathbb{I})]^{-1}(k(X^tX)^{-1}) \\ &= k(X^tX + k\mathbb{I})^{-1}(X^tX)(X^tX)^{-1} = k(X^tX + k\mathbb{I})^{-1}\end{aligned}$$

$$thus, \quad \sum_j (\mathbb{E}[\hat{\beta}_j^R] - \beta_j)^2 = \beta^t(\mathbb{I} - Z)^t(\mathbb{I} - Z)\beta = k^2\beta^t(X^tX + k\mathbb{I})^{-2}\beta \tag{1.18}$$

an the sum of their variance,

$$\begin{aligned}\sum_j Var[\hat{\beta}_j^R] &= tr(Cov(\hat{\beta}^R)) = \sigma^2 tr(Z(X^tX)^{-1}Z^t) \\ &= \sigma^2 tr((X^tX)^{-1}Z^tZ) = \sigma^2 tr((X^tX)^{-1}Z^2)\end{aligned}$$

$$thus, \quad \sum_j Var[\hat{\beta}_j^R] = \sigma^2 \sum_j \frac{1}{\lambda_j} \times \frac{\lambda_j^2}{(\lambda_j+k)^2} \tag{1.19}$$

hence,

$$MSE(\hat{\beta}^R, k) = k^2\beta^t(X^tX + k\mathbb{I})^{-2}\beta + \sigma^2 \sum_j \frac{1}{\lambda_j}.\frac{\lambda_j^2}{(\lambda_j+k)^2}$$

Remark 1.3.2. *The function $\lambda_1(k)$ is a monotonic increasing function of k while $\lambda_2(k)$ is monotonically decreasing.*

The constant k reflects the amount of bias increased and the variance reduced. When $k = 0$, it becomes LSE.

Theorem 1.3.2 (*Hoerl and Kennard (1970)*)**.** *There always exists a $k > 0$ such that,*

$$MSE(\hat{\beta}^R, k) < MSE(\hat{\beta}^R, 0) = MSE(\hat{\beta}^{LS})$$

proof. *later in chapter 3, similar the proof of theorem 3.1.2*

in other words, the Ridge estimator can out perform the *LSE* in terms of providing a smaller *MSE*.Nevertheless, in practice, the choice of k is yet to be determined and hence there is no guarantee that a smaller MSE always be attained by Ridge regression.

Before we take and example, it is important to not that ,the Ridge solution is not invariant under scaling of inputs.Thus, one should standardize both the inputs and the response

$$x_{ij}^{'} = \frac{x_{ij} - \bar{x_j}}{s_{x_j}} \quad and \quad y_i^{'} = \frac{y_i - \bar{y}}{s_y} \quad such\ that \quad \frac{1}{n}\sum_i x_{ij}^{'} = 0\ \frac{1}{n}\sum_i y_i^{'} = 0\ and \quad \frac{1}{n}\sum_i x_{ij}^{'2} = 1$$

before computing the shrinkage estimator in (1.12).Besides, the intercept β_0 is automatically suppressed when working with standardized data.

1.3.2 Lasso Regression

The Lasso (Least Absolute Shrinkage and Selection Operator) is another shrinkage method like Ridge regression, yet with an important and attractive feature in variable selection.

Ridge regression does have one obvious disadvantage; unlike *best subset, forward step-wise,backward step-wise*[7], which will generally select models that involve just a subset of variables, Ridge regression will include all p predictors in the final model.The penality $\lambda \parallel \beta \parallel^2$ in (1.14) will shrink all of the coefficients towards zero, but it will not set any of them exactly to zero (unless $\lambda = \infty$).This may not be a problem for prediction accuracy, but it can create a challenge in model interpretation in setting in which the number of variables p is quite large.Increasing the value of λ will tend to reduce the magnitudes of the coefficients, but will not result in exclusion of any of the variables.

The Lasso is a relatively recent alternative to Ridge regression that overcomes this disadvantage .The Lasso estimator of β is obtained by

$$minimizing \left\{ \parallel y - X\beta \parallel_2^2 \right\} \quad subject\ to \sum_{j=1}^{p} |\beta_j| \leq s\ for\ some\ constant\ s \tag{1.20}$$

Namely, the L_2 penality $\parallel \beta \parallel^2 = \sum_{j=1}^{p} \beta_j^2$ in Ridge regression is replaced by the L_1 penality

[7] methods used in low-dimension regression to select the most appropriate variables for a best model

$\| \beta \|_1 = \sum_{j=1}^{p} |\beta_j|$ in Lasso. The Lagrangian problem become:

$$minimize_{\beta \in \mathbb{R}^p} \{ \| y - X\beta \|^2 + \lambda \| \beta \|_1 \}. \tag{1.21}$$

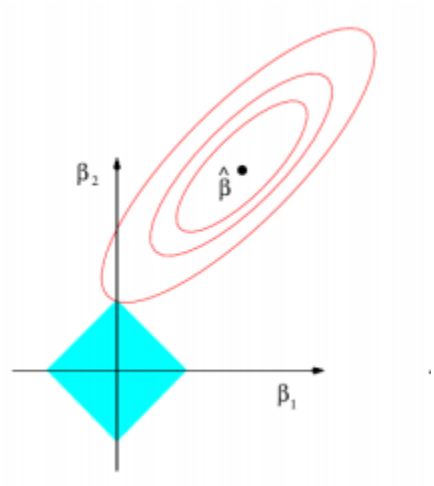

Figure 1.3: contour of the error and constraint function for the Lasso regression in two dimensional case. The solid blue area is the constraint region $|\beta_1| + |\beta_2| \leq s$, while the red ellipse is the contour of RSS. *(figure from.* [14], *P.*244)

Figure 1.3 portrays the Lasso estimation problem in two dimensional case.The constraint region in Ridge regression has a disk shape while the constraint region in Lasso is a diamond. Both methods find the first point at which the elliptical contours hit the constraint region.However, unlike disk, the diamond has corners. If the solution occurs at a corner, then it has one coefficient $\hat{\beta}_j$ equal to zero.L_1 penality has the effect of forcing some of the coefficient estimates to zero when the turning parameter λ is sufficiently large.Hence, much like *best subset selection*, the Lasso performs variable selection.

As a result, model generated from Lasso are generally much easier to interpret than those produced by Ridge regression.we can say that the Lasso yield *sparse models*, that is models that involve only a subset of the variable.

Computation of Lasso Solution

The Lasso problem is a convex program, specifically a quadratic program (**QP**) (*visit [16] for more detail.*) with a convex constraint.As such, there are many sophisticated **QP** methods for solving the Lasso. However, there is a particularly simple an effective computational algorithm, that gives insight into how the Lasso works. The Lagrangian form (1.21)is especially convenient

for numerical computation of the solution.

$$minimize_{\beta\in\mathbb{R}^p}\{\parallel y - X\beta \parallel^2 + \lambda \parallel \beta \parallel_1\} \Leftrightarrow minimize_{\beta\in\mathbb{R}^p}\{\frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}$$

we will assume that both y_i and the features x_{ij} have been standardized (the intercept term β_0 can be omitted).

Let first consider a single predictor setting, based on samples $\{(x_i, y_i)\}_{i=1}^n$ the problem then is to solve $mimizing_{\beta\in\mathbb{R}^p}\{\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta|\}$. The standard approach to this univariate minimization problem would be to take gradient (first derivative) with respect to β, and set it to zero.There is a complication however, because the absolute value function $|\beta|$ does not have a derivative at $\beta = 0$.However, we can proceed by inspection of the function

$$f(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta| = \frac{1}{2n}(\sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n} x_i^2\beta^2 - 2\beta\sum_{i=1}^{n} x_i y_i) + \lambda|\beta|$$

(*recall we assumed that both y_i and the features x_{ij} have been standardized*)

$$= \frac{1}{2n}(n + n\beta^2 - 2\beta < x, y >) + \lambda|\beta| = \frac{1}{2} + \frac{1}{2}\beta^2 - \beta\frac{< x, y >}{n} + \lambda|\beta|$$

Proposition 1.3.4 (*from [21]*)**.** *if β and $\hat{\beta}^{Lasso}$ are the true and Lasso estimator in a single predictor setting based on samples $\{(x_i, y_i)\}_{i=1}^n$,then we have:*

$$\hat{\beta}^{Lasso} = sign(\hat{\beta}^{LS})(|\hat{\beta}^{LS}| - \lambda)_+, \quad (x_+{}^8). \tag{1.22}$$

proof.

$$f(\beta) = \begin{cases} \frac{1}{2} + \frac{1}{2}\beta^2 + \beta(\lambda - \frac{<x,y>}{n}) \; si \; \beta \geq 0 \\ \frac{1}{2} + \frac{1}{2}\beta^2 - \beta(\lambda + \frac{<x,y>}{n}) \; si \; \beta < 0 \end{cases}, \quad f^{'}(\beta) = \begin{cases} \beta + (\lambda - \frac{<x,y>}{n}) \; si \; \beta \geq 0 \\ \beta - (\lambda + \frac{<x,y>}{n}) \; si \; \beta < 0 \end{cases}$$

[8] $x_+ = max(x, 0)$

for $\beta > 0$,

- if $\frac{<x,y>}{n} > \lambda$, $f^{'}(\beta_1) = 0 \Rightarrow \beta_1 = \frac{<x,y>}{n} - \lambda > 0$

β	0	β_1	$+\infty$
$f^{'}(\beta)$	$-$	0	$+$
$f(\beta)$	$f(0)$ ↘	$f(\beta_1)$	↗ $+\infty$

$argmin_\beta f(\beta) = \beta_1$

- if not, say $\frac{<x,y>}{n} \leq \lambda$, $f^{'}(\beta_1) = 0 \Rightarrow \beta_1 = \frac{<x,y>}{n} - \lambda \leq 0$

β	0	$+\infty$
$f^{'}(\beta)$	$+$	
$f(\beta)$	$f(0)$	↗ $+\infty$

$argmin_\beta f(\beta) = 0$

for $\beta < 0$,

- if $\frac{<x,y>}{n} < -\lambda$, $f^{'}(\beta_2) = 0 \Rightarrow \beta_2 = \frac{<x,y>}{n} + \lambda < 0$

β	$-\infty$	β_2	0
$f^{'}(\beta)$	$-$	0	$+$
$f(\beta)$	$+\infty$ ↘	$f(\beta_2)$	↗ $f(0)$

$argmin_\beta f(\beta) = \beta_2$

- if not, say $\frac{<x,y>}{n} \geq -\lambda$, $f^{'}(\beta_2) = 0 \Rightarrow \beta_2 = \frac{<x,y>}{n} - \lambda \geq 0$

β	$-\infty$	0
$f^{'}(\beta)$	$-$	
$f(\beta)$	$+\infty$ ↘	$f(0)$

$argmin_\beta f(\beta) = 0$

we find that,

$$\hat{\beta}^{Lasso} = \begin{cases} \frac{<x,y>}{n} - \lambda & if \ \frac{<x,y>}{n} > \lambda \\ \frac{<x,y>}{n} + \lambda & if \ \frac{<x,y>}{n} < -\lambda \\ 0 & if \ |\frac{<x,y>}{n}| \leq \lambda \end{cases} \tag{1.23}$$

remark that, $\hat{\beta}^{LS} = (x^t x)^{-1} x^t y = (\sum_{i=1}^n x_i^2)^{-1} < x, y >= \frac{<x,y>}{n}$, thus we can succinctly rewrite $\hat{\beta}^{Lasso}$ as :

$$\hat{\beta}^{Lasso} = sign(\frac{< x, y >}{n})(|\frac{< x, y >}{n}| - \lambda)_+ = sign(\hat{\beta}^{LS})(|\hat{\beta}^{LS}| - \lambda)_+.$$

We see that the Lasso shrinks the least squares coefficient toward zero by a constant amount λ; least square coefficient that is less than λ in absolute value is shrunken entirely to zero.The fact that some Lasso coefficients are shrunken entirely to zero explains why the Lasso performs feature selection.

Remark 1.3.3. *Using this intuition from univariate case, we can develop a simple coordinate wise scheme for solving the full Lasso problem* (1.21) *.More precisely, we repeatedly cycle through the predictors in some fixed (but arbitrary) order (say, $j = 1, ..., p$), where at the j^{th} step, we update the coefficient β_j by minimizing the objective function in this coordinate while holding fixed all other coefficients $\{\hat{\beta}_k,\ k \neq j\}$ at their current values.*

Writing the objective function in (1.21) as

$$f(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{1\leq k\neq j\leq p} x_{ik}\beta_k - x_{ij}\beta_j)^2 + \lambda\sum_{j\neq k}|\beta_k| + \lambda|\beta_j|$$

we see that the solution for each β_j can be expressed in terms of the partial residual $r_i^{(j)} = y_i - \sum_{1\leq k\neq j\leq p} x_{ik}\hat{\beta}_k$ as

$$\hat{\beta}_j^{\,Lasso} = sign(\frac{< x_j, r^{(j)} >}{n})(|\frac{< x_j, r^{(j)}}{n}| - \lambda)_+ \,, (1.22) \quad where\, x_j = (x_{1j}, ..., x_{nj})^t,\ r^{(j)} = (r_1^{(j)}, ..., r_n^{(j)})^t$$

equivalently, since the full residual $r_i = \sum_{k=1}^{p} x_{ik}\hat{\beta}_k = r_i^{(j)} - x_{ij}\hat{\beta}_j$, we have $r^{(j)} = r + \hat{\beta}_j x_j$ where $r = (r_1, ..., r_n)^t$ then,

$$\begin{aligned}\frac{< x_j, r^{(j)} >}{n} = \frac{1}{n} < x_j, r + \hat{\beta}_j x_j > &= \frac{1}{n} < x_j, r > + \frac{1}{n}\hat{\beta}_j \underbrace{< x_j, x_j >}_{\frac{1}{n}\sum_i^n x_{ij}^2 = n} \\ &= \frac{1}{n} < x_j, r > + \hat{\beta}_j\end{aligned}$$

therefore, the update can be written as:

$$\hat{\beta}_j^{Lasso} = sign(\frac{1}{n} < x_j, r > + \hat{\beta}_j)(|\frac{1}{n} < x_j, r > + \hat{\beta}_j| - \lambda)_+ \quad for\ all\ j = 1, ..., p. \tag{1.24}$$

Theoretical properties of Lasso penalty

A common assumption of Lasso model is **sparsity**, i.e only a small number of covariates influence the outcome.

Let $S = \{j : \beta_j \neq 0\}$ the index set of non-zero components of the true coefficient vector $\beta \in \mathbb{R}$ and denote the number of relevant covariate by $s = card\{S\}$.Under sparsity assumption, most components of β are zero such that $s \ll p$.For any $\lambda \geqslant 0$ define the active set of the Lasso, $\hat{S}(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0\}$.Given β ,we order the covariates such that $S = \{1, \ldots, s\}$, $S^c = \{s+1, \ldots, p\}$ and considering the partitioning $X = (X_S, X_{S^c})$ where $X_S \in \mathbb{R}^{n\times s}$ contains the n measurements of the s relevant covariates, and $X_{S^c} \in \mathbb{R}^{n\times(p-s)}$ contains the n measurements of the $(p-s)$ irrelevant covariates.Sample covariance matrix are denote by Σ_X and the empirical

covariance is given by $S_{XX} = \frac{X^T X}{n}$.

State the following basic inequality,

Lemma 5. *we have,*

$$\frac{1}{n} \parallel X\hat{\beta}^{Lasso} - X\beta \parallel_2^2 + \lambda \parallel \hat{\beta}^{Lasso} \parallel_1 \leq \frac{2\epsilon^t X(\hat{\beta}^{Lasso} - \beta)}{n} + \lambda \parallel \beta \parallel_1 . \tag{1.25}$$

Proof. refer to $\big[$ [2], $P.103\big]$

The random part part $2\epsilon^t X(\hat{\beta}^{Lasso} - \beta)$ can be bounded in term of the L_1 norm of the parameters involved:

$$2|\epsilon^t X(\hat{\beta}^{Lasso} - \beta)| \leq \big(\max_{1 \leq j \geq p} 2|\epsilon_t X_{(j)}|\big) \parallel \hat{\beta}^{Lasso} - \beta \parallel_1 = 2 \parallel \epsilon^t X \parallel_\infty \parallel \hat{\beta}^{Lasso} - \beta \parallel_1 . \tag{1.26}$$

Now let us introduce the set,

$$\mathcal{A} = \{\frac{2}{n} \parallel \epsilon^t X \parallel_\infty \leq \lambda_o\},$$

for a suitable value of λ_o, the set $\mathcal{A}$ has large probability. Indeed, with Gaussian errors this follow from the following lemma:

Lemma 6. *Suppose that the diagonal elements of the Gram matrix $\frac{X^T X}{n}$ equal 1 for all j.Then we have for all $t > 0$ and for $\lambda_0 = 2\sigma\sqrt{\frac{t^2 + 2\log(p)}{n}}$,*

$$\mathbb{P}(\mathcal{A}) \geq 1 - 2\exp(-\frac{t^2}{2}) \tag{1.27}$$

Proof. see [[2], P.104]

Corollary 1 (Lasso estimation consistency)**.** *Let the assumption of lemma **6** hold. For some $t > 0$, let the regularization parameter be $\lambda = 2\hat{\sigma}\sqrt{\frac{t^2 + 2\log(p)}{n}}$, where $\hat{\sigma}$ is some estimator of σ. Then with probability at least $1 - \alpha$, where $\alpha = 2\exp(-\frac{t^2}{2}) + \mathbb{P}\big(\hat{\sigma} \leq \sigma\big)$.We have:*

$$\frac{2}{n} \parallel X(\hat{\beta}^{Lasso} - \beta) \parallel_2^2 \leq 3\lambda \parallel \beta \parallel_1 \tag{1.28}$$

we thus conclude that, taking the regularisation parameter λ of order $\sqrt{\frac{\log(p)}{n}}$ and assume that $\parallel \beta \parallel_1 = o\big(\sqrt{\frac{n}{\log(p)}}\big)$, result in consistency of the Lasso.

This means that , up to the $\log(p) - term$ and compatibility constant Φ_o^2, the mean squared prediction error is of the same order as if one knew a priori which of the covariates are relevant and using ordinary least squares estimation based on the true relevant s only. See also [*Theorem 14.6, Chap 14 from Guedon et al. (2007)*] for the corresponding result for the random design.

Let us define the vectors β_S and β_{S^c} by:

$$\beta_{j,S} = \beta_j \mathbb{1}_{\{j \in S\}}, \quad \beta_{j,S^c} = \beta_j \mathbb{1}_{\{j \notin S\}}. \tag{1.29}$$

Clearly, $\beta = \beta_S + \beta_{S^c}$; β_S has zeroes outside the index set S and the elements of β_{S^c} can only be non-zero in the complement S^c of S.

Definition 1.3.1 (Compatibility condition)**.** *We say the the compatibility conditionis met for the set S if for some $\Phi_0 > 0$ and for all $\beta \in \mathbb{R}^p$ such that $\| \beta_{S^c} \|_1 \leq 3 \| \beta_S \|_1$, it holds that*

$$\| \beta_S \|_1^2 \leq \frac{1}{n} \frac{s \| X\beta \|_2^2}{\Phi_o 2^2} = \frac{s(\beta^t S_{XX} \beta)}{\Phi_o^2}. \tag{1.30}$$

Theorem 1.3.3. *Suppose the compatibility condition holds for S.Then on $\mathcal{A}$, we have for $\lambda \geq 2\lambda_0$*

$$\frac{1}{n} \| X(\hat{\beta}^{Lasso} - \beta) \|_2^2 + \lambda \| \hat{\beta}^{Lasso} - \beta \|_1 \leq \frac{4\lambda^2 s}{\Phi_o^2}. \tag{1.31}$$

Proof. the proof for this result is clearly detailed in [*[2], Theorem 6.1,P.107*] using the following lemma:

Lemma 7. *On $\mathcal{A}$, with $\lambda \geq 2\lambda_0$ we have:*

$$\frac{2}{n} \| X(\hat{\beta}^{Lasso} - \beta) \|_2^2 + \lambda \| \hat{\beta}_{S^c}^{Lasso} \|_1 \leq 3\lambda \| \hat{\beta}_S^{Lasso} - \beta_S \|_1 . \tag{1.32}$$

Proof. on $\mathcal{A}$, by basic inequality (1.25) and using $\lambda \geq 2\lambda_0$,

$\frac{2}{n} \| X(\hat{\beta}^{Lasso} - \beta) \|_2^2 + 2\lambda \| \hat{\beta}^{Lasso} \|_1 \leq \lambda \| \hat{\beta}^{Lasso} - \beta \|_1 + 2\lambda \| \beta \|_1 .(*)$. Using triangular inequality (second form), $\| \hat{\beta}_S^{Lasso} - \beta_S \|_1 \geq \| \beta_S \|_1 - \| \hat{\beta}_S^{Lasso} \|_1$. Thus ,

$\| \hat{\beta}^{Lasso} \|_1 = \| \hat{\beta}_S^{Lasso} \|_1 + \| \hat{\beta}_{S^c}^{Lasso} \|_1 \geq \| \beta_S \|_1 - \| \hat{\beta}_S^{Lasso} - \beta_S \|_1 + \| \hat{\beta}_{S^c}^{Lasso} \|_1$ $(**)$.in the other hand,
$\| \hat{\beta}^{Lasso} - \beta \|_1 = \| \hat{\beta}_S^{Lasso} - \beta_S \|_1 + \| \hat{\beta}_{S^c}^{Lasso} \|_1$, $(***)$. Using both inequalities $(**)$ and $(***)$ in $(*)$ yield the result.

Remark 1.3.4. *The theorem combines two results:*

$$\frac{2}{n} \| X(\hat{\beta}^{Lasso} - \beta) \|_2^2 \leq \frac{4\lambda^2 s}{\Phi_o^2}, \quad (the\ bound\ for\ predictions\ error) \tag{1.33}$$

$$\| \hat{\beta}^{Lasso} - \beta \|_1 \leq \frac{4\lambda^2 s}{\Phi_o^2}, \quad (the\ bound\ for\ L_1 - error\ of\ coefficients\ estimates.) \tag{1.34}$$

Corollary 2 (estimation accuracy of β)**.** *Under compatibility assumptions on design matrix X and on the sparsity $s = card\{S\}$, for lambda in the suitable range of order $\lambda \approx \sqrt{\frac{log(p)}{n}}$,*

$$\| \hat{\beta}^{Lasso} - \beta \|_1 \xrightarrow[n\to+\infty]{\mathbb{P}} 0 \; ; \quad \| \hat{\beta}^{Lasso} - \beta \|_2 \xrightarrow[n\to+\infty]{\mathbb{P}} 0 \tag{1.35}$$

Proof. see *Knight and Fu (2000)*.

Knowing that Lasso is widely use for model selection, it is necessary to assess how well the sparse model given by Lasso relates to the true model.We make this assessment by investigating Lasso's model consistency (under linear model); That is, for $S = \{j, \beta_j \neq 0\}$ being the true active set, we look for a Lasso procedure delivering an estimator $\hat{S} = \{j, \hat{\beta}_j^{Lasso} \neq 0\}$ of S such that $\hat{S} = S$ with large probability.

Since using Lasso estimate involves choosing the appropriate amount of regularization, to study the model selection consistency of the Lasso, we consider two problems: whether there exists a deterministic amount of regularization that gives consistent selection, or for each random realization whether there exists a correct amount of regularization that selects the true model.The so-called **"irrepresentable condition"** thoroughly interpreted by *Zhao and Yu (2006) [24]* is almost necessary and sufficient for both types of consistency.

An estimate which is consistent in term of parameter estimation does not necessarily consistently select the correct model (or even attempt to do so) where the reverse is also true.The former requires $\hat{\beta}^{Lasso} - \beta \xrightarrow[n\to+\infty]{\mathbb{P}} 0$ while the latter requires $\mathbb{P}\big(\{\hat{S} = S\}\big) \xrightarrow[n\to+\infty]{\mathbb{P}} 1$. We desire our estimate to have both consistencies. However, to separate the selection aspect of consistency from the parameter estimation aspect.We make the following definitions about *"sign*[9] *consistency"* that does not assume the estimates to be estimation consistent.

Definition 1.3.2. *An estimate $\hat{\beta}_n$ is equal in sign with the true model β if and only if,*

$$Sign(\hat{\beta}_n) = Sign(\beta)$$

Definition 1.3.3. *Lasso is strongly sign consistent if there exists $\lambda_n = f(n)$, that is , a function independent of Y and X such that:*

$$\lim_{n\to\infty} \mathbb{P}\big(\{Sign(\hat{\beta}^{Lasso}) = Sign(\beta)\}\big) = 1, \; (*)$$

Definition 1.3.4. *Lasso is general sign consistentcy if*

$$\mathbb{P}\big(\{\exists \lambda \geq 0, Sign(\hat{\beta}^{Lasso}) = Sign(\beta)\}\big) = 1, \; (**)$$

Remark 1.3.5. • *Strong sign consistency implies one can use a preselected λ to achieve consistent*

[9] $Sign(.)$ maps positive entry to 1 ,negative to -1 and 0 to 0

model selection via Lasso.

- *General sign consistency means for a random realization there exists a correct amount of regularization that select true model.*
- $(*) \Rightarrow (**)$

Definition 1.3.5 (Irrepresentable Condition)**.** *We say that , Irrepresentable condition is met for the set S if there exists a constant* $\theta \in [0, 1[$ *such that ,*

$$\| S_{XX}(S^c, S) S_{XX}(S, S)^{-1} sign(\beta_S) \|_\infty \leq \theta.. \tag{1.36}$$

Theorem 1.3.4 (Variables selection consistency)**.** *The irrepresentable condition* (1.36) *for the active set S is a sufficient and essentially necessary condition for Lasso to select only variables in active set S; that is to achieve sign consistency.*

Proof. refer to *Zao and Yu (2006) [24]* or *Meinshausen and Buhlmann (2010)* for more details.

Remark 1.3.6. *The irrepresentable condition , as given in* (1.36) *depends on the Gram matrix* $\frac{X^t X}{n}$ *but also on the signs of the true unknown parameter* β*, whereas the compatibility condition* (1.30) *only depends on* Σ_X*.*

Proposition 1.3.5. *The irrepresentable condition implies the compatibility condition.*

Proof. see [*[2], chap7, P.195*].

Sign consistency is stronger than the usual selection consistency which only requires the zero to be matched , but not signs. It is needed for proving the necessity of the *irrepresentable condition* to avoid dealing with situations where a model is estimated with matching zeros but reversed sign.

1.3.3 Dantzig Selector (DS)

The Lasso is not the only $L_1 - penalization$ possible. from the score equation ,the Dantzig Selector by *Candes and Tao [4]* also belongs to the class of regularisation methods in regression.It can be formulated as the Lasso but instead of controlling the squared error loss, it controls the correlation of residuals with X.Specifically, the Dantzig selector estimator is defined to be the solution of the minimization problem:

$$\min_{\beta \in \mathbb{R}^p} \{ \| \beta \|_1 \} \text{ subject to } \| X^t(y - X\beta) \|_\infty := \sup_{1 \leq i \geq p} |(X^t r)_i| \leq \lambda_p . \sigma, \tag{1.37}$$

for some $\lambda_p > 0$, *where* $r = y - X\beta$ is the residual vector.

The intuition behind the program (1.37) is that, we seek and estimator $\hat{\beta}$ with minimum complexity (as measured by the $L_1 - norm$) among all objects that are consistent with data.

Remark 1.3.7. *The constraint on the residual vector imposes that for each* $j \in \{1, \ldots, p\}$, $|\left(X^t r\right)_j| \leq \lambda_p.\sigma$, *which guarantees that the residuals are within the noise level.*

The Dantzig selector and Lasso are closely related.Connections between the Dantzig Selector and the Lasso have been discussed in *Jame et al. (2008)* where it is shown that under some general conditions, the Dantzig Selector and the Lasso produce the same solution path.

Both models share the feature of setting some of parameters to zero i.e they perform variable selection.

Remark 1.3.8. *Though under some general conditions , the Lasso and Dantzig may produce the same solution path, they differ conceptually in that the Dantzig stems directly from an estimating equation, whereas the Lasso stems from a likelihood or an objective function.*

The theoretical results (estimation accuracy and model selection consistency) for the Dantzig selector estimator are provide with detailed supporting proof in [*[4], theorem 1.1; theorem 1.2*]

1.3.4 Elastic-Net Regression

We ended the section on Lasso regression by saying that it works best when your model contains a lot of useless variables. We also said that Ridge regression works best when most of the variables in your model are useful.

Remark 1.3.9. *When we know about all of the parameters in our model, it's easy to choose if we want to use Lasso or Ridge regression; but what do we do when we are in high dimension setting where the model include tons more variables, far too many to know everything about ?.*

When you have million of parameters, then you will almost certainly need to use some sort of regularization to estimate them.However, the variables in those models might be useful or useless; we don't not in advance.So how do we choose if we should use Lasso or Ridge regression?.

The good news is that we don't have to choose, instead, we use *Elastic-Net* regression. Just like Lasso and Ridge regression, Elastic-Net regression starts with least squares, then it combines the Lasso regression penalty $\lambda_1 \parallel \beta \parallel_1$ with the Ridge regression penalty $\lambda_2 \parallel \beta \parallel_2^2$. The Lagrangian problem become

$$minimize_\beta\{\parallel y - X\beta \parallel^2 + \lambda_1 \parallel \beta \parallel_1 + \lambda_2 \parallel \beta \parallel^2\}.$$

Altogether, Elastic-Net regression combines the strengths of Lasso and Ridge regression. Note that the Lasso and Ridge regression penalty get their own $\lambda's$; λ_1 for Lasso and λ_2 for Ridge.But more often, the problem is writing as

$$minimize_\beta\{\parallel y - X\beta \parallel^2 + \lambda(\alpha \parallel \beta \parallel_1 + (1-\alpha) \parallel \beta \parallel^2)\}, \quad for\ \alpha \in [0,1]\ and\ \lambda \geq 0$$

, say

$$\hat{\beta}^E(\lambda,\alpha) = argmin_\beta\{\| y - X\beta \|^2 + \lambda(\alpha \| \beta \|_1 + (1-\alpha) \| \beta \|^2)\}. \tag{1.38}$$

We still have the regularization parameter λ , but we only have one regularization parameter common to both terms, we also have a parameter α which will control the mix between L_1 and L_2 regularization.

Remark 1.3.10. *We notice that:*

- $\hat{\beta}^E(\lambda,1) = \hat{\beta}^{Lasso}(\lambda)$
- $\hat{\beta}^E(\lambda,0) = \hat{\beta}^R(\lambda)$
- $\hat{\beta}^E(0,\alpha) = \hat{\beta}^{LS}$
- *and when* $\alpha \notin \{0,1\}$ *and* $\lambda \neq 0$ *, then we get the hybrid of Ridge and Lasso estimation.*

Why Elastic-Net ?

The hybrid Elastic-Net regression is especially good at dealing with situations when there are high correlations between parameters.This is because on it's own, Lasso regression tend to pick just one of the correlated terms and eliminates the others, whereas Ridge regression tends to shrink all of the parameters for the correlated variables together; By combining Lasso and Ridge regression, Elastic-Net regression groups and shrinks the parameters associated with the correlated variables and leaves them in equation or removes them all at once.

1.3.5 Selecting The Appropriate Tuning Parameter

Implementing Ridge and Lasso regression requires a method for selecting a value for the tuning parameter λ in (1.14) and (1.21) or equivalently, the value of the constraint s in (1.13) and (1.20).

One way to find a good value of λ is to calculate the MSE of *prediction* (MSE_p) by some sort of *Cross-Validation* for many different values of λ.

The principle is the folloing: we choose a grid of λ values, and compute the Cross-Validation error for each value of λ, as we will describe later.We then select the tuning parameter value for which the Cross-Validation error is smallest.Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

Cross-validation to find the best value of λ

There are various methods to select the "best" value for λ. One is to split the data into **K** chunks. We then use **K-1** of this as a training set, and the remaining 1 chunk as the test set.We can repeat this until we've rotated through all **K** chunks, giving us a good estimate of how well each of the

lambda values work in our data.This is called *cross-validation*, and doing this repeated *test/train* split gives us a better estimate of how generalisable our model is.

We can use this new idea to choose a lambda value, by finding the lambda that minimises the error across each of the test and training splits.

Let (X_k, y_k) denote the subset of X and y for the $k-th$ fold, with $k = 1, \ldots, K$.The optimal λ is obtained by minimizing the total *Cross-validation* error:

$$\hat{\lambda} = \underset{\lambda}{argmin} \underbrace{\left\{ \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \parallel y_k - X_k \hat{\beta}_k(\lambda) \parallel_2^2 \right\}}_{CV_{(K)}}, \tag{1.39}$$

Where n_k is the number of observations in the $k-th$ fold and $\hat{\beta}_k(\lambda)$ is the Lasso (resp. Ridge) estimates based on (X_{-k}, y_{-k}) (the data after removing the $k-th$ fold) and the tuning parameter λ.

A particular case of this method is the so called **Leave-one-out cross-validation (LOOCV)** where **K**=n. In this case, the Cross-validated estimate λ is given by:

$$\hat{\lambda}_n = \underset{\lambda}{argmin} \underbrace{\left\{ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - X_i^t \hat{\beta}_i(\lambda) \right)^2 \right\}}_{CV_{(n)}} \tag{1.40}$$

1.4 Numerical Implementation

We present here two illustrative numerical applications. The first one is based on simulated data and the last one on real data.The purpose of the numerical experiment is to show the behaviour and to investigate if there was an difference in predictive power between the previous three regularization methods; ridge, Lasso and Elastic-net regression when they were applied on high-dimensional data.The statistical analysis was implement using **R** statistical software [**?**].

1.4.1 Simulated Data

For the simulation study, we use generalized linear model (GLMs) for penalized logistic regression.The "*glmnet*" [**?**] package for **R** fits a GLM via penalized maximum likelihood. We will not provide a theory about GLMs in this study ; for specific information regarding GLMs we refer to [**?**] .The measures that are used to assess how good a logistic regression model is for prediction are : *misclassification error rate* (ME) which denotes the fraction of incorrect classifications over all observations and the *Area Under Curve* (AUC) which is a measure of discrimination tanking values between 0 and 1 (visit [**?**] for more details).The simulation study was inspired

by the paper by *Krona* [**?**]. However, adjustments were made to the simulated datasets.

Process description : The simulated data consisted of four independent high-dimensional datasets.Each dataset was divided into a training and a test set.The three methods were used to fit a corresponding model to each of the training sets.The fitted models were used to make predictions for each of the corresponding test sets.Finally, we computed the AUC , the ME and extracted the number of non-zero $\hat{\beta}$-coefficients.The procedure was repeated 100 times per example.

Simulation design : We simulated p=1000 predictor and n=200 observation such that $p >> n$ and the data qualified as high-dimensional. All predictor variables X were continuous multivariate normal distributed except for the binary response variable Y. A multiple group of predictors with varying strength of correlation were simulated for each data set. The predictors were generated by sampling from a multivariate normal distribution with the following probability density function :

$$f_X(x) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det(\Sigma)}} \exp\left\{ -\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu) \right\}$$

were μ is the mean vector and $\Sigma = (\rho_{ij})_{i,j}$ is the covariance matrix.For all x, we set $\mu = 0$ and $Var[x] = 1$ .Thus, Σ equal the correlation matrix of X. Each predictor variable was assigned a predetermined $\beta - value$ .The response variable were simulated by running the simulated data through the inverse logit function (see [**?**]) ,

$$\pi(x) = \frac{1}{1 + e^{-X^t\beta}}.$$

Given the threshold $\pi_0 = 0.5$, the observed value was categorized into one of the two classes $Y = 1$ if $\pi(x) > 0.5$ and $Y = 0$ if $\pi(x) \leq 0.5$ .Consequently, we obtained a vector Y and a matrix X consisting of 200 observations of the binary response variable and the predictor variables respectively.

Details information about the four examples :

Example 1 : we set the pairwise correlation between $X_i \; and \; X_j$ predictors to $\rho_{ij} = 0.5^{|i-j|}$. We assigned the first 122 β-coefficients a specified vector that consisted of random values within [2,5].The remaining coefficients were set to 0.

Example 2 : we set $\rho_{ij} = 0.5^{|i-j|}$. we set all coefficients to be $\beta = 0.8$.

Example 3 : we set $\rho_{ij} = 0.9^{|i-j|}$. The coefficients were split in 8 groups, where the coefficients were set to pairwise be 0 and 2, $\beta = (\underbrace{2,2,\ldots,2}_{125},\underbrace{0,0,\ldots,0}_{125},\underbrace{2,2,\ldots,2}_{125},\ldots)^t$.

Example 4 The pairwise correlation between the first 500 predictors X_i *and* X_j ($1 \leq i,j \leq 500$) were set to $\rho_{ij} = 0.5^{|i-j|}$ and the pairwise correlation for the remaining predictors were set to 0.We set the first 500 coefficients to $\beta = 3$ and the remaining coefficients to 0, $\beta = (\underbrace{3,3,\ldots,3}_{500},\underbrace{0,0,\ldots,0}_{500})^t$.

models	**Example 1**			**Example 2**		
	AUC	ME	Nb. of $\hat{\beta} \neq 0$	AUC	ME	Nb. of $\hat{\beta} \neq 0$
Ridge	0.76 (0.042)	0.32 (0.053)	1000	0.76 (0.050)	0.31 (0.052)	1000
Lasso	0.65 (0.106)	0.41 (0.094)	29	0.55 (0.058)	0.46 (0.056)	20
Elastic Net	0.75 (0.062)	0.37 (0.074)	316	0.70 (0.076)	0.37 (0.068)	329
	Example 3			**Example 4**		
models	AUC	ME	Nb. of $\hat{\beta} \neq 0$	AUC	ME	Nb. of $\hat{\beta} \neq 0$
Ridge	0.92 (0.028)	0.16 (0.041)	1000	0.76 (0.047)	0.31 (0.041)	1000
Lasso	0.84 (0.037)	0.24 (0.048)	54	0.58 (0.073)	0.46 (0.070)	21
Elastic Net	0.90 (0.033)	0.17 (0.045)	415	0.70 (0.059)	0.36 (0.054)	361

Table 1.3: Simulation results.*The table reports the AUC, ME-values and number of non-zero* $\hat{\beta} - coefficients$ *.The simulation was repeated 100 times for each example and all results are reported as median values and (standard deviation sd.)*

Results : The simulation of Example 1-4 was repeated 100 times: for every simulation, we calculated AUC, ME and their standard deviations (sd).In addition, the average number of selected variables by Lasso and the Elastic net was calculated.The results are summarized in table 1.3.

In example 1, a small subset of predictors were assigned non-zero β-coefficients.On average, the Lasso and the elastic net selected 28 and 316 variables respectively. We see that the Ridge regression has the highest AUC and the lowest ME.

In example 2, the predictors were assigned coefficients β=0.8 with relatively high correlation amount predictors.As demonstrate in table 1.3, ridge regression improve over other methods considering AUC and ME. As mention in *subsection 1.3.1*, ridge regression tend to perform well under the circumstances in example 2.Moreover, the average number of coefficients for Lasso and elastic net was 20 and 328 respectively.In this setting, the elastic-net identify a larger number of coefficients that were correlated and non-zero.The Lasso, on the other hand results in a sparse final model but identify less of the non-zero coefficients.instead, the chosen model resulted in a high ME (table 1.3).

In example 3, the predictors were divided into 8 groups and pairwise assigned coefficients of 0 and 2.We see that ridge regression outperform the Lasso and elastic net in view of the AUC.Since the elastic net and ridge regression perform considerably similar, they seem to perform equally as good in this setting.As discussed in earlier, (**subsection 1.3.2**) , ridge regression included all predictors in the final model and resulted in a less interpretable model.However, the elastic net identified on average 415 non-zero coefficients.supposedly, the elastic net adopted the grouping effect and correctly identified almost all non-zero coefficients simultaneously as it achieved high prediction accuracy.

In example 4, the predictors were divided into two groups of equal size that were assigned with β=3 and β=0 respectively.The first 500 were correlated while the remaining 500 predictors were uncorrelated.As seen in table 1.3 , ridge regression achieved the highest AUC while elastic net succeeded to identified approximately all non-zero coefficients as a result of the grouping effect.

Summary The results show that the three methods perform well in the sense that AUC$\geq$0.5 in examples 1-4.We observe that despite the fact that ridge regression tend to spread the coefficients shrinkage over a large number of coefficients, it achieve high predictive power through-out example 1-4.especially, the results in example 3 demonstrated the capacity of ridge regression.We identify that when the number of predictor are very large and a larger fraction of them most be included in the model, ridge regression dominates the Lasso and the elastic net.Consequently, it confirm that ridge regression is satisfactory method for prediction on correlated datasets.The results from example 2 determine that the Lasso is outperformed by the elastic net.Furthermore we observed that the elastic net benefits from the ability to put a larger weight to the quadratic penalty, while it simultaneously shrinks some coefficients to zero by the absolute penalty.

Moreover, we observe that ridge regression and the elastic net generally improve over the Lasso.We can see that elastic net approximately identified all-non zero coefficients in the simulations.In example 4, elastic-net performed grouped selection and showed to be a better variable selection method than Lasso.Even though ridge regression did not incorporate variable selection, it achieved high prediction accuracy through-out example 1-4.Therefore, we observe that if the interpretability is not fundamental, ridge regression manage to accomplish high predictive power.Ultimately, the elastic net has the advantage of incorporating variable selection.Consequently , its final model is more interpretable than that of ridge regression.

Note: full R code are provided in Appendix A.1

1.4.2 Real data example

Data description: For real data example, we will be working with *"human DNA methylation data"* from *"flow-sorted blood samples"* [**?**]. DNA methylation assays measure for each of many sites in the genome, the proportion of DNA that carries a methyl mark (a chemical modification that does not alter the DNA sequence).In this case, the methylation data come in the form of normalised methylation levels (M-values) where negative values correspond to unmethylated DNA and positive values correspond to methylated DNA.Along with this, we have a number of sample phenotypes (e.g BMI, Sex, Age in year).This methylation object is a *"GenomicRatioset"*, a *Bioconductor*[**?**] data object derived from the *"SummarizedExperiment"*[**?**].These *"SummarizedExperiment"* objects contain *"assays"*, in this case normalised methylation levels, and optional sample level *"ColData"* and feature-level *"metadata"*.These objects are very convenient to contain all of the information about a dataset in a high-throughput context.For more details on these objects, one could consult the *vignettes on Bioconductor*....url....

After reading in the data we can see in the provided R output that this object has $dim()\ of\ 5000 \times 37$, meaning it has 5000 features and 37 samples (observations).to extract the matrix of methylation M-values, we use "*assay()" function*.Note that in the matrix of methylation data, samples or observations are stored as rows.

In this episode, we will focus on the association between **Age** and **methylation**.

Experiment steps : Let's denote by X the methylation matrix,

1) Singularity: we investigete singularity of the matrix X^tX and check out what happen if we try to fit linear model to the data.

2) Ordinary least square versus Ridge regression : here, we work with a set of feature known to be associated with **Age** from a paper by *Horvath et al.*.Horvath et al. used methylation markers alone to predict the biological **Age** of an individual.

- we extra the first 20 features of the features identified by Horvath, investigate correlations and we split the methylation data matrix and the age vector into training an test sets.
- we fit both linear regression and ridge regression on the training data matrix and training **Age** vector using the previous features and record the MSE between our predictions and the true **Age**s for the test data.

3) Apply egularization methods : we perform the Lasso, Ridge and Elastic-net on the whole DNA methylation data using cross validation to select the tuning parameter, examine the coefficients paths for each method and load Horvath signature to compare features selected by Lasso and the elastic-net methods.

Results.

1) We can see that we are able to get some effect size estimates, but they seem vert high.The *"Summary"* also says that we were unable to estimate effects sizes for 4964 features because of singularities.What this mean is that R couldn't find a way to perform the calculations necessary due to the fact that we have more features than observations.

2) Preditors are correlated each other.Since we split the data into test and training data, we can see that ridge regression gives us a better prediction on unseen data despite being worse on train data : $MSE_{lm} = 45.14 \geq MSE_{ridge} = 25.30$. *see also comments of Figures 1.4 and 1.5*

3) Comparing the feature selected by Lasso (41 features) and the elastic net (60 features) with Horvath signature, we can see that we selected some of the same feature (8 features for Lasso and 11 features for elastic net).*see also the comments of the remaining 6 figures.*

- **Full R codes are provided in Appendix A.2**

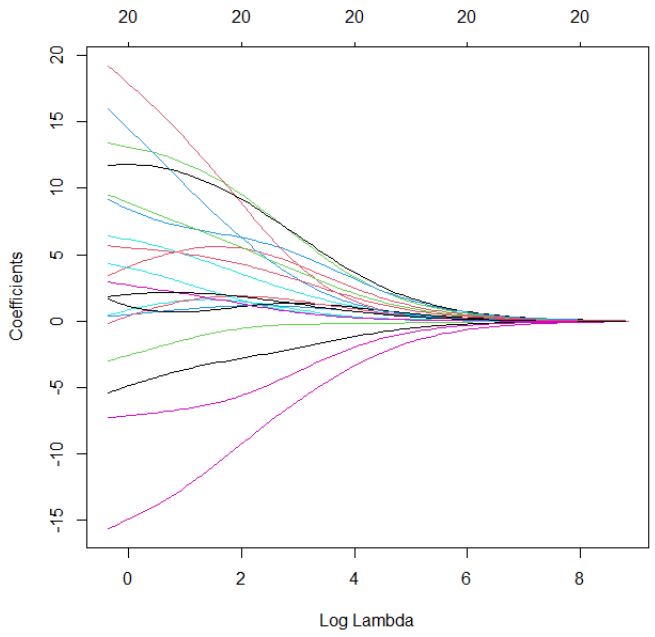

Figure 1.4: *plot showing how estimated coefficients for each methylated site change as we increase the penalty λ. We can see that initially, some parameter estimates are really large, and these tend to shrink fairly rapidly.*

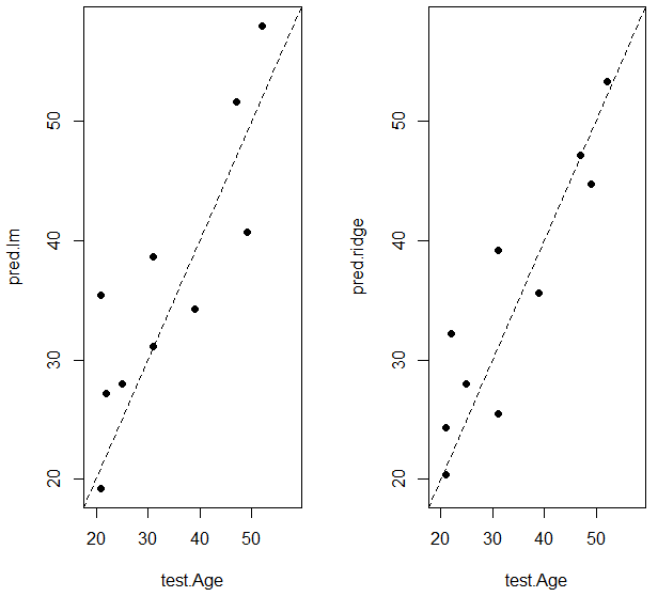

Figure 1.5: *Predicted Ages for each methods against the true Ages. The ridge ones are much less spread out with far fewer extreme predictions.*

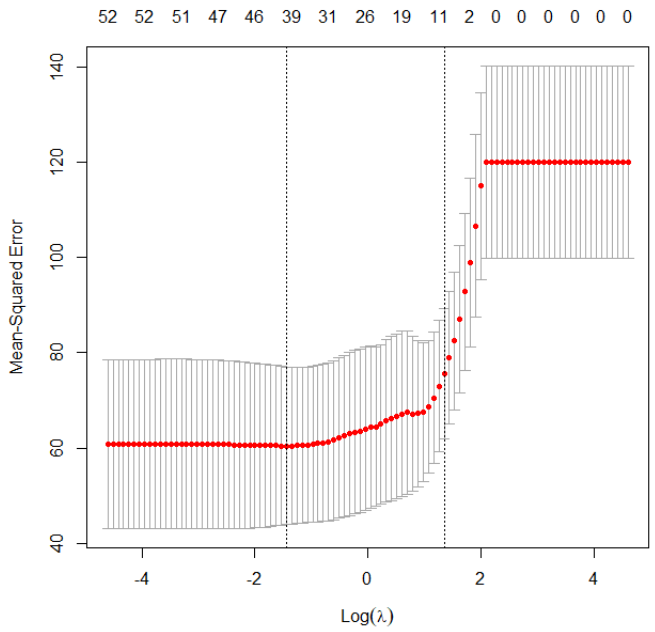

Figure 1.6: *Cross-validation performance for Lasso.*

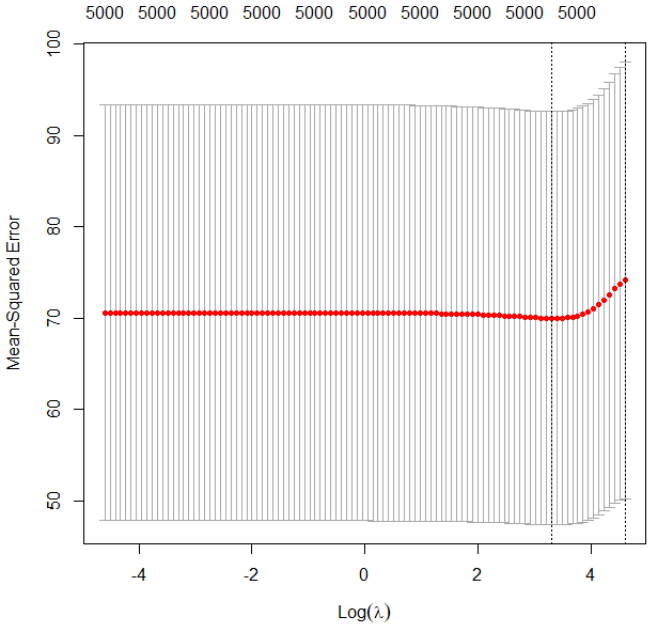

Figure 1.7: *Cross-validation performance for Ridge.*

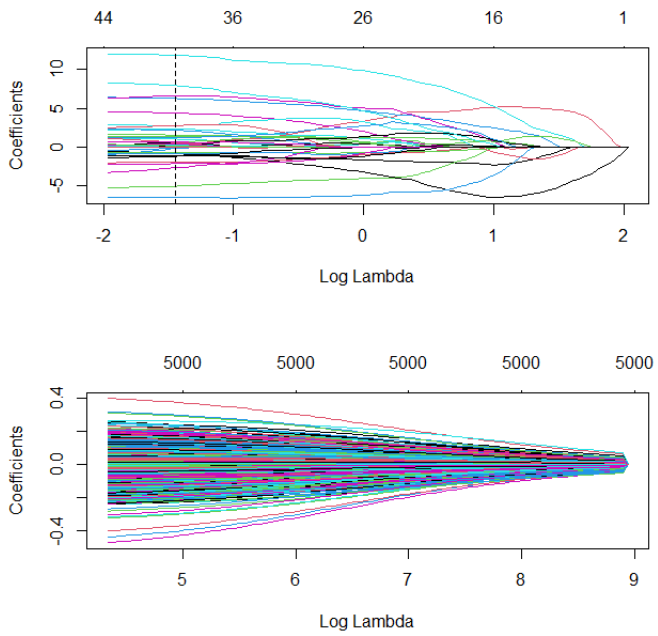

Figure 1.8: *The paths tend to go exactly to zero much more when sparsity increases when we use lasso model.In ridge case, the paths tends toward zero but less commonly reach exactly zero.*

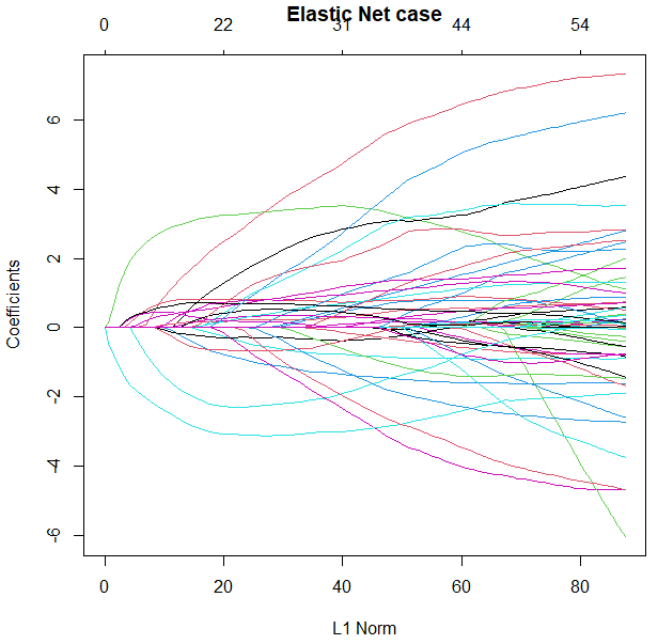

Figure 1.9: *Coefficients paths elastic net.We can see that coefficients tend to go exactly to zero, but the paths are a bit less extreme than with pure Lasso; similar to ridge.*

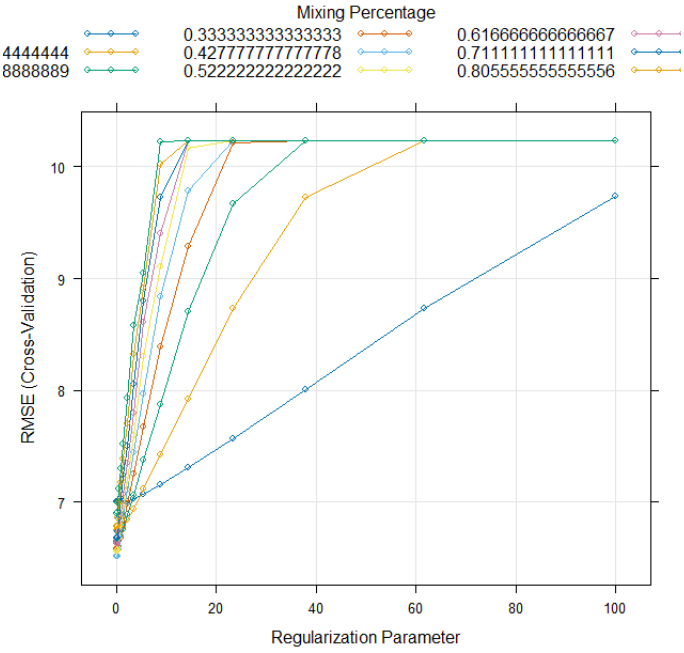

Figure 1.10: *Cross-validation to find the optimal pair of* (α, λ) *for elastic net (mixing percentage).*

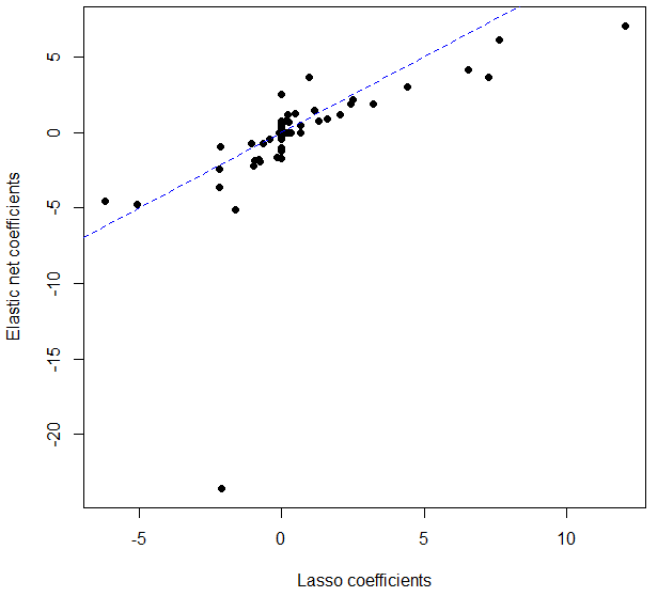

Figure 1.11: *Lasso coefficients against elastic net coefficients.We can see that the coefficients from these two methods are broadly similar, but the elastic net coefficients are a bit more conservative.*

Chapter 2

Measurement Error In Regression theory

2.1 Introduction

2.1.1 Definition an motivating examples

This section is about measurement error in statistical analyses.In some sense, all statistical problems involve measurement error.

Measurement error occur whenever we cannot exactly observe one or more of the variables that enter into a model of interest.There are many reason such errors occur, the most common ones being "sampling error and instrument error".Where any notation is used here, the true value is denoted "X" and the variable observed in place of "X" by "W" (error-prone measurement). When the true and observed values are both categorical, then measurement error is more specifically referred to as **misclassification**.

Measurement error occur in nearly every discipline; Here is a collection of examples in biomedical field:

Genomic: In recent decades, genetic and epigenetic studies have become increasingly more important in medical research, but the process of sequencing DNA typically involves some errors.

Disease statut: I n epidemiology , the outcome variable is often presence or absence of a disease such as breast cancer, hepatitis, AIDS....This is often assessed through an imperfect diagnostic procedure such as an imaging technique or a blood text which can lead to either false positives or false negatives (misclassification).

2.1.2 Objective and some terminology

- how to model measurement error ?
- what the effects of ignoring it are ?

- How, if at all can we correct for measurement error ?

These are three general objective in measurement error problem we will try to address in this parts of our work.

There are typically three main ingredients in measurement error problem:

1) A model for true values: This can be essentially any statistical model.

2) A measurement error model: That is specification of the relationship between the true and observed values.

3) Extra information: Data or assumption that may be needed to correct for measurement error which may not be always available.This extra information is among others:*validation data* in which both true an mismeasured values are obtained on a set of unit; *Replicate values*; *Knowledge about some of the measurement error parameters or fonctions of them*;....

In the sequel, we will focus on the classical additive non-differential measurement error model in the structural case with the assumption of homoskedasticity which refers to the case where the variance of "W" given $X = x$" is constant.

2.2 The Model Description

One of the fundamental assumption in the linear regression analysis is that all observations are correctly observed.When this assumption is violated the measurement error creep into the data.The the usual statistical tools tend to loose their validity(*see [7] and [13] for more details.*). And important issue in the area of measurement errors is to find the consistent estimators of the parameters which can be accomplished by using some additional information from outside the sample.

In section **??** and **??** we consider a linear regression model defined in (1.3) with additive error,

$$y = X\beta + \epsilon \ , \ W = X + U \tag{2.1}$$

$X_i = (X_{i1}, \ldots, X_{ip})^t, \quad W_i = (W_{i1}, \ldots, W_{ip})^t, \quad U_i = (U_{i1}, \ldots, U_{ip})^t;$

$$X = \begin{bmatrix} X_1^t \\ \vdots \\ X_n^t \end{bmatrix} \ n \times p \ matrix; \ U = \begin{bmatrix} U_1^t \\ \vdots \\ U_n^t \end{bmatrix} \ n \times p \ matrix; \ W = \begin{bmatrix} W_1^t \\ \vdots \\ W_n^t \end{bmatrix} \ n \times p \ matrix$$

For the sake of notation simplicity, we assume that $\beta_0 = 0$. The true covariate X are not observed, and instead we have noisy measurements $W = X + U$ where U is and $n \times p$ random

noise matrix with covariance matrix Σ_U.If the $k-th$ variable has been measured correctly, the corresponding column of U will be set equal to zero, as will the variance of the measurement error of the $k-th$ variables, $\Sigma_{U(k,k)}=0$.

Assumption

- the matrix of measurement error $U \in \mathbb{R}^{n\times p}$ is assumed to have normally distributed rows , with mean zero and covariance Σ_U.
- furthermore, assume that ϵ and U are independent and Σ_U is a $p \times p$ matrix of Known values with non-negative diagonal elements.

Remark 2.2.1. *It follow from the structural model*

$$y_i = \beta^t X_i + \epsilon_i,\ W_i = X_i + U_i \tag{2.2}$$

that the vector $\left(y_i, W_i^t\right)^t$ *follows a p+1-variate normal distribution with mean* $\mu = \left(\beta^t\mu_X, \mu_X^t\right)^t$ *and the covariance matrix,*

$$\Gamma = \begin{bmatrix} \sigma_Y^2 & \Sigma_{YW} \\ \Sigma_{WY} & \Sigma_W \end{bmatrix} = \begin{bmatrix} \sigma^2+\beta^t\Sigma_X\beta & \beta^t\Sigma_X \\ \Sigma_X\beta & \Sigma_X+\Sigma_U \end{bmatrix}. \tag{2.3}$$

(easy to verify)

This lead to:

$$y_i|W_i = w_i = \gamma^t w_i + \delta_i \tag{2.4}$$

where$\delta = (\delta_1, \ldots, \delta_n)^t$ are i.i.d normally with mean zero and variance σ^2_{delta}.

Theorem 2.2.1. *Under the given assumptions,* γ *and* σ_δ^2 *are given by,*

$$\gamma = \left(\Sigma_W\right)^{-1}\Sigma_X\beta = \left(\Sigma_X+\Sigma_U\right)^{-1}\Sigma_X\beta \tag{2.5}$$

$$\sigma_\delta^2 = \sigma^2 + \beta^t\Sigma_X\beta - \gamma^t\left(\Sigma_X+\Sigma_U\right)\gamma \tag{2.6}$$

Proof. mimicking what was done in the simple linear case in *Gleser [15].*

Thus

$$\beta = \mathcal{K}_X^{-1}\gamma. \tag{2.7}$$

where $\mathcal{K}_X = \left(\Sigma_X+\Sigma_U\right)^{-1}\Sigma_X$ is a $p \times p$ matrix referred to as the *reliability matrix ,see Gleser (1992) [15] and Aickin and Ritenbaugh (1992)* for example,discussion and illustrations of the role of reliability matrix.

2.2.1 Estimated Coefficients and Behaviour of naive analyses

Statistical analysis that is carried out by ignoring the presence of the measurement error is called a naive approach. Without measurement error, we saw that the estimated coefficients and the unbiased estimator of σ^2 are given by $\hat{\beta} = (X^tX)^{-1}X^ty$ (1.4) and $\hat{\sigma}^2 = \frac{1}{n-p}\sum_i(y - \hat{y}_i)^2$, *with* $\hat{y}_i = \hat{\beta}^t x_i$.

Proposition 2.2.1. *The maximum likelihood estimators of γ and σ_δ^2 are just the naive least squares estimators,*

$$\hat{\beta}_{naive} = \hat{\gamma} = (W^tW)^{-1}W^ty = S_{WW}^{-1}S_{Wy}\ ,\ \hat{\sigma}^2_{naive} = \hat{\sigma}_{delta} = \frac{1}{n-p}\sum_i(y - \hat{y}_i)^2,\ with\ \hat{y}_i = \hat{\beta}^t_{naive}w_i \tag{2.8}$$

where, $S_{WW} = \frac{W^tW}{n}$ is the unbiased estimator of Σ_W and $S_{Wy} = \frac{W^ty}{n}$

Proposition 2.2.2. *The exact bias expression for the naive estimators under the given assumptions are given by:*

$$\mathbb{E}\left[\hat{\beta}_{naive}\right] = \gamma = \mathcal{K}_X\beta\ ,\ \mathbb{E}\left[\hat{\sigma}^2_{naive}\right] = \sigma^2_\delta \tag{2.9}$$

Remark 2.2.2. *This result lead to an important conclusion: The measurement error in one of the variables may induce bias in the estimation of all coefficients including those measured without error.If more covariates are affected by measurement error, the resulting bias may become rather complex and the effect of measurement error may become difficult to describe.*

2.2.2 Correcting for Measurement Error in Multilinear regression

With some exceptions (*see [3], chap11 and 12*), correcting for measurement error requires informations or data as laid out in item **3)** section 2.1.2.

Myriad approaches to carrying out corrections for measurement error have emerged,A number of which are described in [3].These include *direct bias correction, moment based approach, likelihood based techniques,SIMEX and techniques based on modifying equations.*

Proposition 2.2.3. *When Σ_U is known and $\mathcal{K}_X$ is unknown, then $\mathcal{K}_X$ is estimated consistently by replacing Σ_X and $\Sigma_W$by their respective consistent estimators as:*

$$\hat{\Sigma}_X = \hat{\Sigma}_W - \Sigma_U\ ,\ \hat{\Sigma}_W = S_{WW} = \frac{W^tW}{n};\quad and\ we\ have\ \hat{\mathcal{K}}_X = S_{WW}^{-1}\big(S_{WW} - \Sigma_U\big). \tag{2.10}$$

Corollary 3. *The maximum likelihood estimates of β and σ^2 are given by :*

$$\hat{\beta} = \hat{\mathcal{K}}_X^{-1}\hat{\gamma} = \big(S_{WW} - \Sigma_U\big)^{-1}S_{Wy},\ ,\ \hat{\sigma}^2 = \hat{\sigma}^2_\delta - \hat{\beta}^t\Sigma_U\hat{\mathcal{K}}_X\hat{\beta} \tag{2.11}$$

$\hat{\beta}$ is and unbiased estimator and its covariance is given by:

$$\begin{aligned} Cov(\hat{\beta}) =& Cov(\hat{\mathcal{K}}_X^{-1}\hat{\gamma}) = \mathcal{K}_X^{-1} Cov(\hat{\gamma}) \big(\mathcal{K}_X^{-1}\big)^t \\ =& \mathcal{K}_X^{-1} \big(\underbrace{W^t W}_{n\Sigma_W} \big)^{-1} \sigma_\delta \big(\mathcal{K}_X^{-1}\big)^t = \sigma_\delta \big(\underbrace{n\Sigma_X \Sigma_W^{-1} \Sigma_X}_{C} \big)^{-1} = \sigma_\delta C^{-1} \end{aligned}$$

When measurement error is present and Σ_U is not known, it can be estimated through replicated measurements of W.

Proposition 2.2.4. *Suppose on unit i there are $m_i > 1$ replicated values $W_{i1}, \ldots, W_{im_i}$ of the error-prone measure of x and $\bar{W}_{i.} = \sum_{k=1}^{m_i} \frac{W_{ik}}{m_i}$ their mean. replication allows us to estimate Σ_U as:*

$$\hat{\Sigma}_U = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{k=1}^{m_i} \big(W_{ik} - \bar{W}_{i.}\big)\big(W_{ik} - \bar{W}_{i.}\big)^t}{m_i - 1} \tag{2.12}$$

In that case;

$$\hat{\Sigma}_X = S_{WW} - \hat{\Sigma}_U,\ \hat{\mathcal{K}}_X = S_{WW}^{-1}(S_{WW} - \hat{\Sigma}_U),\ \textit{and}\ \hat{\beta} = \big(S_{WW} - \hat{\Sigma}_U\big)^{-1} S_{Wy} \tag{2.13}$$

Remark 2.2.3. *With sufficiently large measurement error, it is possible that $S_{WW} - \hat{\Sigma}_U$ can be negative.In that case, some adjustment must be made; see Block and Peterson (1975).*

Our discussion of the linear model is intended only to set the stage for our main topic, **measurement error in high-dimensional context** and is far from complete;A vast literature exists on measurement error.There is a number of excellent books, starting with one by *Fuller [13]* who wrote the first influential book focusing on linear regression models, and on by *Caroll et al. [6]* who treated measurement error in a much broader application context.Another book that give wide treatment to the topic is by *Buanaccorsi [7]* who focuses on different topics from those in the aforementioned two books and places emphasis on more applied approach.

Chapter 3

Measurement Error in High-Dimensional Context :Behaviour and Correction Methods

3.1 Ridge Regression Estimation Over Measurement Error Ridden Data.

The standard assumption in the linear regression analysis is that explanatory variables are uncorrelated.When this assumption is violated, the explanatory variables are nearly dependent, which refers as **multicollinearity problem** (very common in high dimensional data) and yields poor estimators of interest parameters as described in section 1.2.1 item *iii)*.In order to resolve this problem, several approaches have been considered among them, the "Ridge regression" introduced by *Horel and Kennard [17]* was discuss in section 1.3.1 and considers a shrinkage method to overcome the problem of multicollinearity for the estimation of regression parameters.

When the problem of multicollinearity is present in the measurement error ridden data , then and important issues is how to obtain the consistent estimators of regression coefficients.One simple idea is to use the ridge regression estimation over the error ridden data.An obvious question that crops up is what happen then?.

In this section, we attempt to answer such questions.

3.1.1 Ridge Regression Estimator of β and its Asymptotic Properties.

Here we introduce the ridge regression estimators of β.For this, we first consider the conditional setup of the least squares method 2.1 with known *reliability matrix* $\mathcal{K}_X$.Remember in this case that the corrected moment estimator or corrected score estimated of β and γ are respectively given by :

$$\hat{\beta}_{ME}^{LS} = \mathcal{K}_X^{-1}\hat{\gamma} = \left(S_{WW} - \Sigma_U\right)^{-1} S_{Wy} \text{ (2.11), } and \ \gamma = \mathcal{K}_X\beta \text{ (2.5)}$$

where "ME" stands for measurement error. The suggested estimator of β based on a shrinkage strategy is obtain by minimizing ,

$$\underset{\beta\in\mathbb{R}^p}{minimize}\left\{ \parallel y - W\gamma \parallel_2^2 \right\} \quad subject\ to\ \parallel \beta \parallel^2 \leq s\ for\ some\ constant\ s \tag{3.1}$$

the Lagrangian problem become

$$\underset{\beta\in\mathbb{R}^p}{minimize}\left\{ \parallel y - W\mathcal{K}_X\beta \parallel_2^2 + k \parallel \beta \parallel^2 \right\} \tag{3.2}$$

Proposition 3.1.1. *The numerical solution of this problem corresponding to the ridge regression estimator of β in measurement error model 2.1 is given by:*

$$\hat{\beta}_{ME}^{R} = \left[\mathbb{I}_p + k\big(n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big)^{-1}\right]^{-1}\hat{\beta}_{ME}^{LS}. \tag{3.3}$$

Proof. set $f(\beta) = \parallel y - W\mathcal{K}_X\beta \parallel_2^2 + k \parallel \beta \parallel^2 = \big(y - W\mathcal{K}_X\beta\big)^t\big(y - W\mathcal{K}_X\beta\big) + k\beta^t\beta$.Analogously to proof of proposition **1.3.2**, by taking partial derivative with respect to each component of β, we obtain $\frac{\partial f(\beta)}{\partial\beta} = 2k\beta + 2\mathcal{K}_X^t W^t\big(y - W\mathcal{K}_X\beta\big)$.

$$\begin{aligned}
\frac{\partial f(\beta)}{\partial\beta} = 0 \Rightarrow & k\beta + \mathcal{K}_X^t W^t W\mathcal{K}_X\beta = \mathcal{K}_X^t W^t y\ i.e\ \big[k\mathbb{I}_p + n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big]\beta = n\mathcal{K}_X^t S_{Wy} \\
\Rightarrow \beta &= \left[\mathbb{I}_p + k\big(n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big)^{-1}\right]^{-1}\big(n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big)^{-1} n\mathcal{K}_X^t S_{Wy} \\
&= \left[\mathbb{I}_p + k\big(n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big)^{-1}\right]^{-1}\mathcal{K}_X^{-1}\underbrace{S_{WW}^{-1}S_{Wy}}_{\hat{\gamma}} = \left[\mathbb{I}_p + k\big(n\mathcal{K}_X^t S_{WW}\mathcal{K}_X\big)^{-1}\right]^{-1}\underbrace{\mathcal{K}_X^{-1}\hat{\gamma}}_{\hat{\beta}_{ME}^{LS}}.
\end{aligned}$$

Corollary 4. *Substituting the consistent estimator of $\mathcal{K}_X$ given in* (2.10) *we get,*

$$\hat{\beta}_{ME}^{R} = \left[\mathbb{I}_p + k\big(n\hat{\mathcal{K}}_X^t S_{WW}\hat{\mathcal{K}}_X\big)^{-1}\right]^{-1}\hat{\beta}_{ME}^{LS}. \tag{3.4}$$

Denote the ridge factor of ridge estimation by: $Z_n^{ME} = \big[\mathbb{I}_p + kC_n^{-1}\big]^{-1}$ with $C_n = n\hat{\mathcal{K}}_X^t S_{WW}\hat{\mathcal{K}}_X$.

Remark 3.1.1. *Z_n^{ME} is a consistent estimator of $Z^{ME} = \big[\mathbb{I}_p + kC^{-1}\big]^{-1}$, with $C = n\mathcal{K}_X^t\Sigma_W\mathcal{K}$ and $\hat{\beta}_{ME}^{R} = Z_n^{ME}\hat{\beta}_{ME}^{LS}$.*

The expectation,covariance and sum of their square bias of $\hat{\beta}_{ME}^{R}$ are given by:

- $\mathbb{E}\big[\hat{\beta}_{ME}^{R}\big] = \mathbb{E}\big[Z_n^{ME}\hat{\beta}_{ME}^{LS}\big] = Z_n^{ME}\mathbb{E}\big[\hat{\beta}_{ME}^{LS}\big] = Z_n^{ME}\beta$.

- $Cov(\hat{\beta}^R_{ME}) = Cov(Z_n^{ME}\hat{\beta}^{LS}_{ME}) = Z_n^{ME}Cov(\hat{\beta}^{LS}_{ME})Z_n^{ME^t} = \sigma_\delta^2 Z_n^{ME}C_n^{-1}(Z_n^{ME})^t$.
- $\sum_{j=1}^p \left(\mathbb{E}[\hat{\beta}^R_{ME_j} - \beta_j]\right)^2 = \beta^t(\mathbb{I}_p - Z_n^{ME})^t(\mathbb{I}_p - Z_n^{ME})\beta = k^2\beta^t[C_n + k\mathbb{I}_p]^{-2}\beta$; the proof is similar to the one of **Proposition 1.3.3** replacing X^tX by C_n.

Corollary 5. *The mean square error of $\hat{\beta}^R_{ME}$ is given by:*

$$MSE(\hat{\beta}^R_{ME}, k) = k^2\beta^t[C_n + k\mathbb{I}_p]^{-2}\beta + \sigma_\delta^2 tr(Z_n^{ME}C_n^{-1}(Z_n^{ME})^t) \tag{3.5}$$

Proof.

$$\begin{aligned} MSE(\hat{\beta}^R_{ME}, k) =& \sum_{j=1}^p \left(\mathbb{E}[\hat{\beta}^R_{ME_j} - \beta_j]\right)^2 + \sum_{j=1}^p Var(\hat{\beta}^R_{ME}) \quad \textit{by } \textbf{Proposition 1.3.1} \\ =& k^2\beta^t[C_n + k\mathbb{I}_p]^{-2}\beta + \underbrace{tr(Cov(\hat{\beta}^R_{ME}))}_{\sigma_\delta^2 tr\left(Z_n^{ME}C_n^{-1}(Z_n^{ME})^t\right)} \end{aligned}$$

Remark 3.1.2.

- *When $n \to \infty$ then $C_n \to C$, $Z_n^{ME} \to Z^{ME}$ and*

$$MSE(\hat{\beta}^R_{ME}, k) = k^2\beta^t[C + k\mathbb{I}_p]^{-2}\beta + \sigma_\delta^2 tr(Z^{ME}C^{-1}(Z^{ME})^t)$$

- *if $k = 0$ then $Z^{ME} = \mathbb{I}_p$ and $MSE(\hat{\beta}^R_{ME}, k) = \sigma_\delta^2 tr(C^{-1}) = MSE(\hat{\beta}^{LS}_{ME})$.*

Comparison of $\hat{\beta}^R_{ME}$ and β^{LS}_{ME}

Let $\lambda_{max} = \lambda_1 \geq \cdots \geq \lambda_p = \lambda_{min} > 0$ denote the eigenvalues of the positive definite matrix $C = n\mathcal{K}_X^t\Sigma_W\mathcal{K}$.we can find and orthogonal matrix P such that, $P^tCP = D = diag(\lambda_1, \ldots, \lambda_p)$ (see **Remark 1.2.1**); The corresponding eigenvalues of Z^{ME} and $[C + k\mathbb{I}_p]^{-1}$ are respectively, $\frac{\lambda_j}{\lambda_j+k}$, $\frac{1}{\lambda_j+k}$ $j = 1, \ldots, p$ so that.

$$k^2\beta^t[C_n + k\mathbb{I}_p]^{-2}\beta = k^2\beta^tP^t[D + k\mathbb{I}_p]^{-2}P\beta = k^2\sum_{j=1}^p \frac{\alpha_j^2}{(\lambda_j + k)^2}, \textit{ where } \alpha = P\beta, \ (p \times 1 vector)$$

and

$$\sigma_\delta^2 tr(Z^{ME}C^{-1}(Z^{ME})^t) = \sigma_\delta^2\sum_{j=1}^p \frac{\lambda_j}{(\lambda_j + k)^2}, \textit{ see } \textbf{Remark 1.2.1 and (1.19)}$$

. Now the MSE of $\hat{\beta}^R_{ME}$ may be written as:

$$MSE(\hat{\beta}^R_{ME}, k) = k^2\sum_{j=1}^p \frac{\alpha_j^2}{(\lambda_j + k)^2} + \sigma_\delta^2\sum_{j=1}^p \frac{\lambda_j}{(\lambda_j + k)^2} = \psi_b(k) + \psi_v(k). \tag{3.6}$$

Theorem 3.1.1. *Under the given assumption,*

i) *The quadratic bias, $\psi_b(k)$ is a continuous, monotonically increasing function of k.*

ii) *$\psi_b(k)$ approach $\beta^t\beta$ as an upper limit*

iii) *As $k \to 0^+$, $\frac{d\psi_b(k)}{dk} \to 0$*

iv) *The total variance $\psi_v(k)$ is a continuous monotonically decreasing function of k.*

Proof.

(i) clearly, $\psi_0(k) = 0$, thus $\psi_b(k)$ is a continuous function of k and $\frac{\psi_b(k)}{dk} = 2k\sum_{j=1}^{p}\sum_{j=1}^{p}\frac{\lambda_j\alpha_j^2}{(\lambda_j+k)^2} > 0$, is non negative hence the result.

(ii) $\lim\limits_{k\to+\infty}\psi_b(k) = \lim\limits_{k\to+\infty}\sum_{j=1}^{p}\frac{\alpha_j^2}{(1+\frac{\lambda_j}{k})^2} = \sum_j\alpha_j^2 = \alpha^t\alpha = \beta^tP^tP\beta = \beta^t\beta$.

- (iii) and (iv) are also easy to verify.

Theorem 3.1.2 (*from [19]*)**.** *There always exist a $k > 0$ such that ,*

$$MSE\big(\hat{\beta}_{ME}^{R}, k\big) < MSE\big(\hat{\beta}_{ME}^{LS}\big) . \tag{3.7}$$

Proof. consider the derivative of $MSE\big(\hat{\beta}_{ME}^{R}, k\big)$ with respect to k, $\frac{MSE\left(\hat{\beta}_{ME}^{R},k\right)}{dk} = \frac{\psi_b(k)}{dk} + \frac{\psi_v(k)}{dk} = 2\sum_{j=1}^{p}\frac{\lambda_j}{(\lambda_j+k)^3}(k\alpha_j^2 - \sigma^2)$ $(*)$. A sufficient condition for $(*)$ to be negative is that, $(k\alpha_{max}^2 - \sigma^2) < 0 \Rightarrow k < \frac{\sigma^2}{\alpha_{max}^2}$ with $\alpha_{max} = \max(\alpha_j)$. Thus, for $0 < k < \frac{\sigma^2}{\alpha_{max}^2}$, $MSE\big(\hat{\beta}_{ME}^{R}, k\big)$ decrease i.e

$$MSE\big(\hat{\beta}_{ME}^{R}, k\big) < MSE\big(\hat{\beta}_{ME}^{R}, 0\big) = MSE\big(\hat{\beta}_{ME}^{LS}\big) \ \textit{as required.}$$

3.2 Measurement Error In Lasso

Modern statistics is facing problems due to the increase of dimensionality of the data in field such as genomics,finance,network analysis,....It is quite canonical in high-dimensional regression, where the number of variables p largely exceeds the sample size n to assume that the number of covariates s that has an effect on the response variable y is much less than n (*sparsity assumption*).Hence, the vector of regression parameters is assumed to be $s-sparse$.A plethora of high-dimensional regression methods is available, among which the "Lasso regression [21] we presented in **subsection 1.3.2**, "Dantzig selector (DS) [7] and Smoothly Clipped Absolute Deviation (SCAD) [13].These methods all allow model selection and parameter estimation through a penalization of the parameters as seen for the Lasso case (**subsection 1.3.2**).These methods

are developed for the case in which the covariates are fully observed and without errors; However, in many applications, our data are subject to at least some measurement error.In classical regression context, when $p < n$ and standard methods can be applied, it is well known that measurement error in the covariates will lead to bias in the estimation of the parameters (2.9) and to loss of power [6].

Since the standard Lasso is widely used despite the present of measurement error, it is of interest to study the effects measurement error has on the analysis and describes some of the statistical methods used to correct for those effects.

3.2.1 Impact Of Ignoring Measurement Error

The notation in **1.3.2 P.21** (used to study proprieties of lasso) is used for W and U.We partition the variance matrix in the form:

$$S_{WW} = \begin{bmatrix} S_{WW}(S,S) & S_{WW}(S,S^c) \\ S_{WW}(S^c,S) & S_{WW}(S^c,S^c) \end{bmatrix} \tag{3.8}$$

We saw that in the absence of measurement error, the Lasso is consistent for prediction and estimation (**Theorem 1.3.3** (1.31)). $y = X\beta + \epsilon = (W+U)\beta + \epsilon = W\beta + \underbrace{\epsilon - U\beta}_{\delta}$.

Proposition 3.2.1. *Assume the compatibility condition* (1.30) *holds with constant* Φ *, and that there exist a constant* λ_0 *such that* $\frac{2}{n} \parallel \delta^t W \parallel_\infty \leq \lambda_0$*; Then, with a regularization parameter* $\lambda \geq 2\lambda_0$*,*

$$\frac{1}{n} \parallel W(\hat{\beta}^{Lasso} - \beta) \parallel_2^2 + \lambda \parallel \hat{\beta}^{Lasso} - \beta \parallel_1 \leq \frac{4\lambda^2 s}{\Phi_o^2}. \tag{3.9}$$

Proof. This is from **Theorem 1.3.3**

This shows that in the presence of measurement error, the estimation error of Lasso can be bounded.Using the triangle inequality, we have

$$\begin{aligned} \parallel \delta^t W \parallel_\infty &= \parallel (\epsilon - U\beta)^t W \parallel_\infty = \parallel \epsilon^t W - \beta^t U^t (X+U) \parallel_\infty \\ &\leq \parallel \epsilon^t W \parallel_\infty + \parallel \beta^t U^t X \parallel_\infty + \parallel \beta^t U^t U \parallel_\infty \leq \parallel \epsilon^t W \parallel_\infty + \parallel \beta^t U^t X \parallel_\infty + \parallel \beta \parallel_1 \parallel U^t U \parallel_\infty \end{aligned}$$

Hence the bound (3.9) is implied by ,

$$\frac{2}{n} \parallel \epsilon^t W \parallel_\infty + \frac{2}{n} \parallel \beta^t U^t X \parallel_\infty + 2 \parallel \beta \parallel_1 \parallel \frac{U^t U}{n} \parallel_\infty \leq \lambda_0 \, ; \tag{3.10}$$

and the Lasso with measurement error is consistent if all the three terms in the above expression

(3.10) converge to 0.However,

$$\frac{U^tU}{n} \underset{n\to+\infty}{\longrightarrow} \Sigma_U \ \ and \ \ \| \Sigma_U \|_\infty \neq 0$$

, consequently, we do not obtain consistency.

We have just see that standard results for consistency of estimation no longer hold when the covariates are affected by measurement error.Now let's see how measurement error affect covariate selection with Lasso.By **Definition 1.3.5** (1.36), the "irrepresentable condition with measurement error (**IC-ME**) hold if there exists a constant $\theta \in [0, 1[$ such that ,

$$\| S_{WW}(S^c, S)S_{WW}(S, S)^{-1} sign(\beta_S) \|_\infty \leq \theta.. \tag{3.11}$$

In presence of measurement error, *Sorensen, Frigessi and Thoren (2015) [20]* shown that to achieve covariate selection consistency, we need the following additional condition called "Measurement Error Condition" (**MEC**):

Definition 3.2.1 (MEC)**.** *The measurement error condition (MEC) is satisfied if*

$$\Sigma_W(S^c, S)\Sigma_W(S, S)^{-1}\Sigma_U(S, S) - \Sigma_U(S^c, S) = 0. \ , \ (visit \ [20] \ for \ more \ details). \tag{3.12}$$

Remark 3.2.1. *The MEC applies to population covariance matrix,whereas the IC-ME applies to sample covariance matrix.*

3.2.2 Correction for Measurement Error in Lasso

The purpose of this section es to describe some penalized regressions correction methods that may be used to correct both the variable selection and the model estimation at the same time assuming measurement error is adequately modelled (in our case " additive measurement error").

To show the bias in the estimation caused by measurement error, consider the naive Lasso approach, plugging in W for X in the Lasso estimator defined in (1.21)

$$\hat{\beta}^{LS}(\lambda_n) = argmin_{\beta\in\mathbb{R}^p}\left\{ \| y - W\beta \|_2^2 + \lambda_n \| \beta \|_1 \right\}. \tag{3.13}$$

It is possible to demonstrate that this yield the bias loss function:

$$\mathbb{E}\left[\| y - W\beta \|_2^2 \, | X, y\right] = \| y - X\beta \|_2^2 + n\beta^t\Sigma_U\beta. \tag{3.14}$$

Indeed, using the properties of the conditional expectation , we have:

$$\begin{aligned}\mathbb{E}\big[\parallel y - W\beta \parallel_2^2 |X,y\big] =& \mathbb{E}\big[(y-W\beta)^t(y-W\beta)\big] = \mathbb{E}\Big[\big((y-X\beta)^t-(U\beta)^t\big)\big((y-X\beta)-(U\beta)\big)\Big] \\ =& \mathbb{E}\big[(y-X\beta)^t(y-X\beta)|X,y\big] - (y-X\beta)^t\beta\underbrace{\mathbb{E}[U]}_{0} - \beta^t(y-X\beta)\underbrace{\mathbb{E}[U^t]}_{0} + \mathbb{E}\big[\beta^tU^tU\beta\big] \\ =& \parallel y - X\beta \parallel_2^2 + \beta^t\mathbb{E}\big[U^tU\big]\beta = \parallel y - X\beta \parallel_2^2 + n\beta^t\Sigma_U\beta \ \ (\textit{remember } \mathbb{E}[\frac{U^tU}{n}] = \Sigma_U).\end{aligned}$$

3.2.3 Corrected Lasso (Non Convex Lasso)

The must natural way for correcting for the bias in (3.14) leads to the constrained correct Lasso (CCL):

$$\hat{\beta}_{CCL} \in \underset{\beta:\|\beta\|_1\leq R}{argmin}\left\{\frac{1}{n}\parallel y - W\beta \parallel_2^2 - \beta^t\Sigma_U\beta\right\}. \tag{3.15}$$

or alternatively , the regularized version (regularize corrected Lasso),

$$\hat{\beta}_{RCL} \in \underset{\beta\in\mathbb{R}^p}{argmin}\left\{\frac{1}{n}\parallel y - W\beta \parallel_2^2 - \beta^t\Sigma_U\beta + \lambda_{RCL}\parallel \beta \parallel_1 \right\}. \tag{3.16}$$

both introduced by *Loh and Wainright (2012) [20]*.

Since in practice we may not know the covariance matrix Σ_X, given the set of samples, it is natural to form the estimates of the quantities Σ_X and $\Sigma_X\beta$ as:

$$\hat{\Sigma_X} = \frac{W^tW}{n} - \Sigma_U \textit{ , and } \hat{\gamma} = \frac{1}{n}W^ty$$

. Notice that Σ_U is in practice unknown and must be estimated from data, (see **Proposition 2.1.4**).

Proposition 3.2.2. *The estimator* (3.15) *and* (3.16) *can be reformulated as:*

$$\hat{\beta}_{CCL} \in \underset{\beta:\|\beta\|_1\leq R}{argmin}\left\{\frac{1}{2}\beta^t\hat{\Sigma}_X\beta - \hat{\gamma}^t\beta\right\}. \textit{ , and} \tag{3.17}$$

$$\hat{\beta}_{RCL} \in \underset{\beta:\|\beta\|_1\leq b_0\sqrt{s}}{argmin}\left\{\frac{1}{2}\beta^t\hat{\Sigma}_X\beta - \hat{\gamma}^t\beta + \lambda_{RCL}\parallel \beta \parallel_1 \right\}. \textit{ , for some constant } b_0. \tag{3.18}$$

Proof. set $f(\beta) = \frac{1}{n} \parallel y - W\beta \parallel_2^2 - \beta^t \Sigma_U \beta$. The result follow from the fact that,

$$
\begin{aligned}
f(\beta) &= \frac{1}{n}(y - W\beta)^t(y - W\beta) - \beta^t \Sigma_U \beta \\
&= \frac{1}{n} y^t y - \frac{1}{n} y^t W\beta - \frac{1}{n}\beta^t W^t y + \beta^t \frac{W^t W}{n}\beta - \beta^t \Sigma_U \beta \\
&= \frac{1}{n} \parallel y \parallel_2^2 - 2\frac{1}{n} y^t W\beta + \beta^t\big(\hat{\Sigma}_X + \Sigma_U\big) - \beta^t \Sigma_U \beta = \frac{1}{n} \parallel y \parallel_2^2 - 2\hat{\gamma}^t\beta + \beta^t \hat{\Sigma}_X \beta \\
\underset{\beta}{argmin}\{f(\beta)\} &= \underset{\beta}{argmin}\left\{\frac{1}{2}\big(f(\beta) - \frac{1}{n} \parallel y \parallel_2^2\big)\right\} = \underset{\beta}{argmin}\left\{\frac{1}{2}\beta^t \hat{\Sigma}_X \beta - \hat{\gamma}^t\beta\right\}
\end{aligned}
$$

Remark 3.2.2. *When $\Sigma_U = 0_{\mathbb{R}^{p\times p}}$ (corresponding to the noiseless case), the estimators reduce to the standard Lasso.However, when $\Sigma_U \neq 0_{\mathbb{R}^{p\times p}}$, the matrix $\hat{\Sigma}_X$ is not positive semidefinite in high-dimensional regime ($p >> n$).Indeed, since the matrix $\frac{1}{n}W^tW$ has rank at must n, the subtracted matrix Σ_U may cause $\hat{\Sigma}_X$ to have a large number of negative eigenvalues.Consequently the quadratic losses appearing in the problems* (3.15) *and* (3.16) *are* ***non convex****.*

Remark 3.2.3. *When, $\hat{\Sigma}_X$ has negative eigenvalues (which happen very often under high-dimensionality), the objective function in equation* (3.16) *is unbounded from below, hence we make use of the regularized estimator* (3.18) *to overcome these technical difficulties.*

Remark 3.2.4. *Note that,"$\in$" and not "=" has been used because in the presence of non-convexity, it is generally impossible to provide a polynomial-time algorithm that converges to a (near) global optimum due to the presence of local minima.*

*Loh and Wainwright [11] demonstrated that a simple "**project gradient descent algorithm**" applied to the problems* (3.15) *or* (3.18) *(if b_0 is properly chosen) converge with high probability to a small neighbourhood of the set of all global minimizers.*

Definition 3.2.2. *Project gradient descent is a standard way to solve constrained optimization problem.*

The correct Lasso has been shown to yield good estimation bounds,see [*[20], Theorem 1 and Theorem 2*]. *Sorensen et al. [20]* derived its asymptotic selection consistency properties.

3.2.4 Convex Conditional Lasso

A clear drawback of the previous method is that it leads to a non-convex optimization problem.The ideal behind CoCoLasso is to intervene directly on $\hat{\Sigma}_X$, the estimated covariance matrix of X, with a transformation that will provide a "positive semi-definite" matrix.

We first introduce some necessary notations and model setup:

- For any square matrix $G = (g_{ij})_{i,j}$, we write $G > 0$ (≥ 0) when it is positive (semi-) definite.

- Let $\| G \|_{\max} = \max_{i,j} |g_{ij}|$ denote the element-wise maximum norm.
- We assume that all variables are centred so the the intercept term is not included in the model.

We now define a nearest positive semi-definite matrix operator as follows:

For any square matrix G,

$$\left(G\right)_+ = \underset{G_1 \geq 0}{argmin} \| G - G_1 \|_{\max} \tag{3.19}$$

This operator will project the matrix $\hat{\Sigma}_X$ into a space of semi-definite matrix selecting the nearest one. Then, by denoting $\widetilde{\Sigma}_X = \left(\hat{\Sigma}_X\right)_+$, the convex conditional Lasso is define as:

$$\hat{\beta}_{CoCo} = \underset{\beta \in \mathbb{R}^p}{argmin} \left\{ \frac{1}{2} \beta^t \widetilde{\Sigma}_X \beta - \hat{\gamma}^t \beta + \lambda_{CoCo} \| \beta \|_1 \right\} \tag{3.20}$$

Remark 3.2.5. *The matrix $\widetilde{\Sigma}_X$ is always positive semi-definite by construction while $\hat{\Sigma}_X$ is guaranteed to be positive semi-definite only for $p < n$.Consequently, the optimization problem in* (3.20) *is guaranteed to be convex.*

Theorem 3.2.1 (Cholesky decomposition)**.** *Let A be a real-valued symmetric (semi-) positive-definite matrix; There exist a lower triangular matrix L with real and positive diagonal entries, such that,*

$$A = L^T L \tag{3.21}$$

Defining $\frac{1}{\sqrt{n}} \widetilde{X}$ the Cholesky factor of $\widetilde{\Sigma}_X$ (i.e $\frac{1}{n} \widetilde{X}^t \widetilde{X} = \widetilde{\Sigma}_X$) and $\widetilde{y}$ such that $\frac{1}{n} \widetilde{X}^t \widetilde{y} = \hat{\gamma} = \frac{1}{n} W^t y$, the estimator (3.20) can be reformulates as:

$$\hat{\beta}_{CoCo} = \underset{\beta \in \mathbb{R}^p}{argmin} \left\{ \frac{1}{n} \| \widetilde{y} - \widetilde{X} \beta \|_2^2 + \lambda_{CoCo} \| \beta \|_1 \right\} \tag{3.22}$$

Remark 3.2.6. *This is a regular Lasso regression of $\widetilde{y}$ and $\widetilde{X}$ with penalization parameter λ_{CoCo} (**Section 1.3.2**).It is of great advantage for the practical implementation.We can apply any standard Lasso algorithm as the coordinate descent algorithm [12] or Least angle regression [10] to obtain solution.*

Theoretical Properties (Consistency assessment)

Theoretically, (3.20) can be analysed by the tools for analysing the clean Lasso. By definition of $\widetilde{X}$, we have:

$$\| \widetilde{\Sigma}_X - \Sigma_X \|_{\max} \leq \| \widetilde{\Sigma}_X - \hat{\Sigma}_X \|_{\max} + \| \hat{\Sigma}_X - \Sigma_X \|_{\max} \leq 2 \| \hat{\Sigma}_X - \Sigma_X \|_{\max} \tag{3.23}$$

Remark 3.2.7. *Equation* (3.23) *ensure that* $\widetilde{\Sigma}_X$ *approximates the true Gram matrix* Σ_X *as well as the initial surrogate* $\hat{\Sigma}_X$ *chosen by Loh and Wanwright which is often an unbiased estimate of* Σ_X *achieving a desired rate of convergence under the max norm* $\| \, . \, \|_{\max}$.

Definition 3.2.3 (Closeness Condition)**.** *Let us assume that the distribution of* $\hat{\Sigma}_X$ *and* $\hat{\gamma}$ *are identified by a set of parameters* θ*.Then, there exist universal constants* C *and* c *and positive functions* ζ *and* ϵ_0 *depending on* β_S, θ *and* σ^2*, such that for every* $\epsilon \leq \epsilon_0$, $\hat{\Sigma}_X$ *and* $\hat{\gamma}$ *satisfy the following probability statements:*

$$\mathbb{P}\left(\left\{ |\hat{\Sigma}_{X_{ij}} - \Sigma_{X_{ij}}| \geq \epsilon \right\} \right) \leq C \exp\left(- nc\epsilon^2 \zeta^{-1} \right) \tag{3.24}$$

$$\mathbb{P}\left(\left\{ |\hat{\gamma}_j - \gamma_j| \geq \epsilon \right\} \right) \leq C \exp\left(- ncs^{-2}\epsilon^2 \zeta^{-1} \right) , \quad \forall i, j = 1, \ldots, p. \tag{3.25}$$

Remark 3.2.8. *This condition required that the surrogates* $\hat{\Sigma}_X$ *(and hence* $\widetilde{\Sigma}_X$ *) and* $\hat{\gamma}$ *are close to* Σ_X *and* γ *respectively in term of the element-wise maximum norm.*

Definition 3.2.4 (Restricted eigenvalue Condition)**.**

$$\min_{v \neq 0_{\mathbb{R}^p}, \| v_{S^c} \|_1 \leq 3 \| v_S \|_1} \left\{ \frac{v^t \Sigma_X v}{\| \, v \, \|_2^2} \right\} = \phi > 0 \tag{3.26}$$

refer to [22] for more details about this condition. We have the following result on the L_1 and L_2 statistical error of the CoCoLasso estimate.

Theorem 3.2.2 (*from [8]*)**.** *Under the assumptions stated in* (3.24),(3.25),(3.26) *; for* $\lambda \leq min\big(\epsilon_0, 12\epsilon_0 \, \| \, \beta_S \, \|_\infty \big)$ *and* $\epsilon \leq min\big(\epsilon_0, \frac{\phi}{64s}\big)$ *the following results holds true with probability at least* $1 - p^2 C \exp(-ncs^{-2}\lambda^2\zeta^{-1}) - p^2 C \exp(-nc\epsilon^2\zeta^{-1})$ *:*

$$\| \, \hat{\beta}_{CoCo} - \beta \, \|_2 \leq C\lambda \frac{\sqrt{s}}{\phi} \; ; \; \| \, \hat{\beta}_{CoCo} - \beta \, \|_1 \leq C\lambda \frac{s}{\phi} \tag{3.27}$$

Proof. proof are provided in [*[8], Section 8, P.14*].

In order to establish the sign consistency of CoCoLasso, In addition to the *Closeness Condition*(3.24) and (3.25), let's assume the *"minimum eigenvalue condition"* and the *irrepresentable condition* which are sufficient and nearly necessary for sign consistency of the standard Lasso(**Theorem 1.3.4**):

$$\| \, \Sigma_X(S^c, S)\Sigma_X^{-1}(S, S) sign(\beta_S) \, \|_\infty < 1 \; , \; \lambda_{min}\big(\Sigma_X(S, S)\big) = C_{min} > 0 \tag{3.28}$$

where $\lambda_{min}\big(\Sigma_X(S, S)\big)$ denote the minimum eigenvalue of $\Sigma_X(S, S)$.

Theorem 3.2.3. *under the assumptions given in equations* (3.24),(3.25) *and* (3.28), *for lambda* $\leq$ $min\big(\epsilon_0, \frac{4\epsilon}{\gamma}\big)$ *and* $\epsilon \leq min\big(\epsilon_1, \frac{\lambda}{(\lambda\epsilon_2+\epsilon_3)}\big)$ *where* ϵ_i'*s are bounded positive constants depending of* $\Sigma_X(S,S)$, β_S, θ *and* γ. *The following occur with probability at least* $1-\psi$ *, with* $\psi = p^2C\exp\big(-ncs^{-2}\lambda^2\zeta^{-1}\big) - p^2C\exp\big(-ncs^{-2}\epsilon^2\zeta^{-1}\big)$ *:*

- *There exist a unique solution* $\hat{\beta}^{CoCo}$ *minimizing* (3.20) *whose support is a subset of the true support.*
- $\parallel \hat{\beta}_S^{CoCo} - \beta_S \parallel_\infty \leq \kappa\lambda$ *where* $\kappa = \big(4 \parallel \Sigma_X^{-1}(S,S) \parallel_\infty + C_{min}^{-\frac{1}{2}}\big)$
- *If* $|\beta_{min}| \geq \kappa\lambda$, *then* $Sign\big(hat\beta_S^{CoCo}\big) = Sign\big(\beta_S$.

Proof. visit [[8], *Theorem 2, Section 8, P.14*] for more details.

Remark 3.2.9. *If we assume for simplicity that* κ *is* $o(1)$ *and the triplet* $\{n,p,s\}$ *and* β *satisfy the scaling:*

$$\frac{s^2\log(p)}{n} \longrightarrow 0 \ \text{as } n,p \to +\infty\ ,\ |\beta_{min}| >> s\left(\zeta\frac{\log(p)}{n}\right)^{\frac{1}{2}}. \tag{3.29}$$

Then from the expression of psi in ***Theorem 2.3.3*** *above, we can choose* λ *so that* $1-\psi$ *goes to one, which implies the sign-consistency of the CoCoLasso estimate.*

Corollary 6 (Sign-Consistency)**.** *If* Σ_X , $\widetilde{\Sigma}_X$ *and* $\hat{\gamma}$ *satisfy the regularity conditions given in* ***Theorem 2.3.3****, Then under the scaling in equation* (3.29)*, the CoCoLasso estimate* $\hat{\beta}^{CoCo}$ *defined in* (3.20) *is sign-consistent if* $|\beta_{min}| >> \lambda >> s\big(\zeta\frac{\log(p)}{n}\big)^{\frac{1}{2}}$ *and we have bound*

$$\mathbb{P}\Big(\parallel \hat{\beta}_S^{CoCo} - \beta_S \parallel_\infty \leq \kappa\lambda\Big) \underset{n\to+\infty}{\longrightarrow} 1 \tag{3.30}$$

Remark 3.2.10. *Unlike asymptotic selection consistency properties for non convex Lasso (NCL) , the which was derived only for restrictive case of additive measurement error, the result provided in this subsection those note requires any specification of the type of measurement error.*

3.2.5 Selecting The Tuning Parameter Under Measurement Error

The choose of the tuning parameter in penalized methods relies on *Cross-Validation* (**Subsection 1.3.5**). In presence of measurement error, naive application of Cross-validation might lead to bias results.To elucidate, consider the usual K-folds Cross-validation for selecting optimal λ in the clean Lasso (1.39).

If we naively use the observed data (W,y) , then the cross-validated choice of λ is defined by minimizing ,

$$CV_{(K)} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\parallel y_k - W_k\hat{\beta}_k(\lambda)\parallel_2^2. \tag{3.31}$$

Even if we use CoCoLasso or NCL to compute $\hat{\beta}_k(\lambda)$ based on $W_{-k} and\ y_{-k}$, the above criterion is biased compared to (1.39) in the same way we shown that the loss function in (3.13) is a biased version of the one in (1.21).Observing that (1.39) is equivalent to:

$$\hat{\lambda} = \underset{\lambda}{argmin}\left\{ \frac{1}{K}\sum_{k=1}^{K} \frac{1}{2}\hat{\beta}_k^t(\lambda)\Sigma_k\hat{\beta}_k(\lambda) - \gamma_k^t\hat{\beta}_k(\lambda) \right\}. \tag{3.32}$$

where $\Sigma_k = \frac{1}{n_k}X_k^tX_k$ and $\gamma_k = \frac{1}{n_k}X_k^ty_k$;see proof of **Proposition 2.3.2** (setting $W = X_k$ and $\beta^t\Sigma_U\beta = 0$).

Since unbiased the unbiased surrogate $\hat{\Sigma}_k$ possibly has negative eigenvalues, using it will lead to a cross validation function unbounded from below.*Datta and Zou [8]* substituted Σ_k and γ_k with their projected and estimated counterparts $\widetilde{\Sigma}_k = \left(\hat{\Sigma}_k\right)_+$ and $\hat{\gamma}_k$ .With this correction, the cross-validated λ is defined as:

$$\tilde{\lambda} = \underset{\lambda}{argmin}\left\{ \frac{1}{K}\sum_{k=1}^{K} \frac{1}{2}\hat{\beta}_k^t(\lambda)\widetilde{\Sigma}_k\hat{\beta}_k(\lambda) - \hat{\gamma}_k^t\hat{\beta}_k(\lambda) \right\}. \tag{3.33}$$

$\tilde{\lambda}$ is an unbiased estimator of λ.More theoretical result about cross-validation under measurement error could be found in [9].

3.3 Matrix uncertainty selector (MU-Selector)

So far, we saw that corrected Lasso (NCL) (3.18) and CoCoLassso correct for measurement error, by including in the model the covariance of the measurement error Σ_U, and yielding estimators with good theoretical properties.However, this quantity is assumed to be known and in practice it is usually not known.The estimation of the covariance matrix of the measurement error requires additional data as replicated measurement of the covariates, and can be computationally expensive or even unfeasible when the number of variables p increases.

An interesting alternative is the so-called *Matrix Uncertainty Selector* proposed by *Rosenbaum and Tsybakov [18]*.

We consider the model in (2.1).We typically assume that β is "s-sparse" where $1 \leq s \leq p$ is some integer. In what follows, we assume that ϵ and U satisfy the assumptions:

$$\frac{1}{n} \parallel W^t\epsilon \parallel_\infty \leq \lambda \ \ and \ \ \parallel U \parallel_\infty \leq \delta. \quad (\ with\ high\ probability\ ). \tag{3.34}$$

The "Matrix Uncertainty Selector" $\hat{\beta}_{MUS}$ is define as the solution of the minimization problem:

$$\min\left\{ \parallel \beta \parallel_1 : \beta \in \Theta, \frac{1}{n} \parallel W^t(y - W\beta) \parallel_\infty \leq (1+\delta)\delta \parallel \beta \parallel_1 + \lambda \right\}, \tag{3.35}$$

where $\Theta \subseteq \mathbb{R}^p$ is a given set characterizing the prior knowledge about β.

The problem (3.35) is a convex minimization problem and it reduces to linear programming if $\Theta = \mathbb{R}^p$ .Throughout this section, we assume for simplicity that all diagonal elements of the Gram matrix $\frac{1}{n}X^tX$ are equal to 1.

Proposition 3.3.1 (solution existence)**.** *Under assumptions* (3.34) *, the feasible set of the convex problem* (3.35) *is non empty,*

$$\Psi = \left\{ \beta \in \Theta, \frac{1}{n} \parallel W^t(y - W\beta) \parallel_\infty \leq (1+\delta)\delta \parallel \beta \parallel_1 + \lambda \right\} \neq \emptyset \tag{3.36}$$

Proof. let's show that the true β^* belong to Ψ. $\beta^* \in \Theta$ and,

$$\begin{aligned} \frac{1}{n} \parallel W^t(y - W\beta^*) \parallel_\infty &= \parallel W^t(X\beta^* + \epsilon - W\beta^*) \parallel_\infty = \parallel W^t\big(\epsilon - (\underbrace{W - X}_{U})\beta^*\big) \parallel_\infty \\ &\leq \frac{1}{n} \parallel W^t\epsilon \parallel_\infty + \frac{1}{n} \parallel W^tU\beta^* \parallel_\infty \leq \lambda + \frac{1}{n} \parallel W^tU\beta^* \parallel_\infty \leq, \text{ by (3.34)} \end{aligned}$$

next by second inequality of (3.34) and by the fact that all the diagonal elements of $\frac{1}{n}X^tX$ are equal to 1, the columns matrix $W_{(j)}$ of W satisfy $\parallel W_{(j)} \parallel_2 \leq \sqrt{n}(1+\delta)$, therefore, we obtain

$$\begin{aligned} \frac{1}{n} \parallel W^tU\beta^* \parallel_\infty &\leq \frac{1+\delta}{\sqrt{n}} \parallel U\beta^* \parallel_2 \leq (1+\delta) \parallel U\beta^* \parallel_\infty \\ &\leq (1+\delta) \parallel U \parallel_\infty \parallel \beta^* \parallel_1 \leq (1+\delta)\delta \parallel \beta^* \parallel_1 . \end{aligned}$$

Hence $\frac{1}{n} \parallel W^t(y - W\beta) \parallel_\infty \leq \lambda + (1+\delta)\delta \parallel \beta \parallel_1$, thus there always exists a solution $\hat{beta}_{MUS}$ of (3.35).

Remark 3.3.1. *If $\delta = 0$ and $\Theta = \mathbb{R}^p$, the MU-Selector becomes the Dantzig selector* (1.37)*.*

The MU-Selector can be seen as an evolution of the Dantzig selector that can also take into account the measurement error in the model without needing any information about the measurement error variance, but rather by using a supplementary tuning parameter ("δ").

Theoretical Properties

For a vector $\beta \in \mathbb{R}^p$ and a subset S of $\{1, \ldots, p\}$, we denote by β_S the vector in $\mathbb{R}^p$ that has same coordinates as β on the set indices S and zero coordinates on its complement S^c ($\beta = \beta_S + \beta_{S^c}$) .

Assume that the matrix X satisfy one of the following conditions (assumptions):

Restricted eigenvalue assumption RE(s): there exist $\phi > 0$ such that :

$$\min_{V \neq 0_{\mathbb{R}^p}, \|V_{S^c}\|_1 \leq \|V_S\|_1} \left\{ \frac{\parallel XV \parallel_2}{\sqrt{n} \parallel V \parallel_2} \right\} \geq \phi \tag{3.37}$$

for all subsets S of $\{1,\dots,p\}$ of cardinality $|S| \leq s$. Detailed discussion of this assumption can be found in [1].

Coherence Condition: all the diagonal elements of the matrix $C = \frac{1}{n}X^tX$ are equal to 1 and its off-diagonal elements $c_{ij}\ i \neq j$, satisfy the coherence condition:

$$\max_{i\neq j}|c_{ij}| \leq \rho \quad \textit{with some } \rho < 1. \tag{3.38}$$

Theorem 3.3.1 (*from [18]*)**.** *Assume that model* (2.1) *holds with an unknown s-sparse parameter vector* β *and that all the diagonal elements of* $\frac{1}{n}X^tX$ *are equal to 1.Let* (3.34) *holds, set* $\nu = 2(2+\delta)\delta \parallel \beta \parallel_1 +2\lambda$. *Then for any solution* $\hat{\beta}_{MUS}$ *of* (3.35)*, we have the following inequalities:*

i) *under restricted eigenvalue assumption RE(s)* (3.37)*,*

$$\parallel \hat{\beta}_{MUS} - \beta \parallel_1 \leq \frac{4\nu s}{\phi^2} \quad , \parallel X(\hat{\beta}_{MUS} - \beta) \parallel_2^2 \leq \frac{4\nu^2 s}{\phi^2} \tag{3.39}$$

ii) *Under Coherence condition assumption* (3.38) *with* $\rho < \frac{1}{3\alpha s}$, $\alpha > 1$*:*

$$\parallel \hat{\beta}_{MUS} - \beta \parallel_\infty \leq \frac{3\alpha+1}{3(\alpha-1)}\nu \tag{3.40}$$

Sketch of proof. set $V = \hat{\beta}_{MUS} - \beta$ and S the set of non zero coordinates of β.we have,
$\parallel U^tX \parallel_\infty = \max\limits_{1\leq j,k\leq p}|U_{(j)}^tX_{(k)} \leq \max\limits_{1\leq j,k\leq p} \parallel U_{(j)} \parallel_2 \parallel X_{(j)} \parallel_2 \leq \delta n \ (*)$
where $U_{(j)}$, $X_{(j)}$ are the column of U and X respectively and we used that, $\parallel X_{(k)} \parallel_2 = \sqrt{n}$ by assumption on $\frac{1}{n}X^tX$, and $\parallel U_{(j)} \parallel_2 \leq \delta\sqrt{n}$ by (3.34).Now, note that (3.34), $(*)$ and the fact that $\hat{\beta}_{MUS}$ belong to the feasible set Ψ (3.36) of (3.35) leads to

$$\parallel \frac{1}{n}X^tXV \parallel_\infty \ (**)$$

Taking into account $(**)$, the proof of (3.39) follows the same lines ass the proof of *[[18], Theorem 7.1]* setting $r = \frac{\nu}{2}$ and $m = s$.
To prove (3.40), one could refer to *[[18], Theorem 1]*.

Remark 3.3.2. *The bounds of* ***2.4.1*** *do not depend on* $\parallel \hat{\beta}_{MUS} \parallel_1$ *but on the unknown* $\beta \parallel_1$ *. This drawback is corrected for small values of* $delta$ *, as shown in the next result.*

Theorem 3.3.2 (estimation consistency)**.** *Let the assumptions of* ***Theorem 2.4.1*** *hold and* $\delta < \frac{\phi^2}{4s}$ *.Set* $\nu_1 = 2(2+\delta)\delta \parallel \hat{\beta}_{MUS} \parallel_1 +2\lambda$ *. then for any solution* $\hat{\beta}_{MUS}$ *of* (3.35)*, we have:*

i) *under restricted eigenvalue assumption RE(s)* (3.37),

$$\| \hat{\beta}_{MUS} - \beta \|_1 \leq \frac{4\nu_1 s}{\phi^2}\left(1 - \frac{4\delta s}{\phi^2}\right)^{-1} \quad , \| X(\hat{\beta}_{MUS} - \beta) \|_2^2 \leq \frac{4\nu_1^2 s}{\phi^2}\left(1 - \frac{4\delta s}{\phi^2}\right)^{-1} \tag{3.41}$$

ii) *Under Coherence condition assumption* (3.38) *with* $\rho < \frac{1}{3\alpha s}$, $\alpha > 1$ *and* $\delta \leq \frac{\phi^2}{8s}$*:*

$$\| \hat{\beta}_{MUS} - \beta \|_\infty \leq \frac{2(3\alpha+1)}{3(\alpha-1)}\nu_1 \tag{3.42}$$

Proof. It goes along the same lines as the proof of *Theorem 2.4.1 , cf. [[18], Theorem 4,P.16].*

Definition 3.3.1. *the threshold estimator* $\widetilde{\beta} = (\widetilde{\beta}_1, ..., \widetilde{\beta}_p)^t$ *is define by:*

$$\widetilde{\beta}_j = \hat{\beta}_j^{MUS} \mathbb{1}_{\left\{|\hat{\beta}_j^{MUS}| > \tau\right\}} \ , \ j = 1, \ldots, p. \tag{3.43}$$

where the threshold is given either by

$$\tau = \frac{2(3\alpha+1)}{3(\alpha-1)}\left(2\lambda + 2(2+\delta)\delta a\right) \ , \ for\ \alpha > 0,\ a > 0. \tag{3.44}$$

or by

$$\tau = \frac{2(3\alpha+1)}{3(\alpha-1)}\left(2\lambda + 2(2+\delta)\delta \| \hat{\beta}_{MUS} \|_1 \right) \ , \ for\ \alpha > 0, \ . \tag{3.45}$$

Theorem 3.3.3 (Selection Consistency)**.** *Let the assumptions of* ***Theorem 2.4.1*** *hold.Let either* $\Theta \subseteq \left\{\beta \in \mathbb{R}^p : \| \beta \|_1 \leq a\right\}$ *for some* $a > 0$ *and the threshold* τ *is given by* (3.44) *or* $\frac{\phi^2}{8s}$ *and the threshold* τ *is given by* (3.45)*.If* $\min_{j \in S} |\beta_j| > \tau$ *, then,*

$$Sign(\widetilde{\beta}_j) = Sign(\beta_j) \ , \ for\ all\ \widetilde{\beta}_j \in (3.43) \ , \ j = 1, ..., p \tag{3.46}$$

Proof. *Cf. [18], Theorem 5,P.17*

In summary, under some assumptions, $\hat{\beta}_{MUS}$ recover β with high accuracy in different norm as well as under prediction risk (**Theorem 2.4.2**); And under somewhat stronger assumptions these estimators recover correctly the sparsity pattern (**Theorem 2.4.3**).

3.4 Numerical Studies

3.4.1 Ridge under measurement error (simulation)

As discuss earlier, ridge regression (1.14) provide better estimators when facing problem of multicollinearity in our data.The purpose of this simulation is to evaluate the performance of the modified ridge estimation in (3.2) when problem of multicollinearity is present in the measurement error ridden data.To this end, we will restrict particularly to the case where p<n (low-dimensional data) with high correlations between covariates measured with error.

Simulation design: We simulate data from the true model ,

$$y = X\beta + \epsilon \quad , \epsilon \rightsquigarrow \mathcal{N}(0,1) \ , \ p = 100 \ and \ n = 500$$

where X has been generated as $X \rightsquigarrow \mathcal{N}\big(0, \Sigma_X\big)$ with $\Sigma_X = (\rho_{ij})$ $(\rho_{ij} = 0.9^{|i-j|})$. All coefficients are set to 3, $\beta = (3, 3, \ldots, 3)^t$.The observed data were generated as ,

$$W = X + U, \quad where \ U \rightsquigarrow \mathcal{N}\big(0, \Sigma_U\big) \ with \ \Sigma_U = 0.75\mathbb{I}_p$$

. The simulated data was divided into a training and a test set.The four methods ;*True OLS*[1] *($y \sim X$)*(1.4) , *corrected OLS*(2.11) , *naive ridge* and *modified ridge regression* (3.2) were used to fit a corresponding model to the training set. The fitted models were used to make predictions to the test set and we computed the MSE and the PE (prediction error on the test set).The procedure was repeated 100 times.

Simulation results: We can see in table 3.1 that both the MSE and PE (on average) of the estimates $\hat{\beta}$ provided by the modified (corrected) ridge are lower than those of the three others.Meaning that the provided $\hat{\beta}$ is much more reliable considering MSE (as mentioned in *theorem 3.1.2*) and PE .We also find out in passing that using the corrected version of OLS (2.11) in this setting (*"high-correlation with measurement error"*) would result to a pretty poor estimator given the MSE and PE (table 3.1). **R codes are provided in appendix A.3**

	true OLS	**corrected OLS**	**naive Ridge**	**corrected Ridge**
MSE	6.85 (0.328)	477.94 (4700)	6.84 (0.327)	6.83 (0.327)
PE	2.01(0.146)	50058. (499396)	0.05 (0.008)	0.04 (0.006)

Table 3.1: *Simulation results for ridge under measurement error. The table reports the PE and the estimation error as l_2 norm (MSE).results are reported as median values and (standard deviation sd.)*

[1]Ordinary least square

3.4.2 Measurement error with sparsity assumption (simulations): NCL, Co-CoLasso and MUS implementation.

Appendix A

R codes

A.1 R code for numerical experiment of Lasso, Ridge and Elastic net.

```
# loading required packages

library(glmnet); library(pROC); library(caret)
# setup
N <- 200 # number of observations
n.sim <- 100 # number of simulations
n.cov <- 1000 # number of covariates

############ four examples of high dimensional data sets ##########
# container matrix  (for misclassifications errors)
lasso.me <- matrix(NA,nrow = n.sim , ncol = 4)
ridge.me <- matrix(NA,nrow = n.sim , ncol = 4)
elnet.me <- matrix(NA,nrow = n.sim , ncol = 4)
#container matrix (for AUC values)
lasso.auc <- matrix(NA,nrow = n.sim , ncol = 4)
ridge.auc <- matrix(NA,nrow = n.sim , ncol = 4)
elnet.auc <- matrix(NA,nrow = n.sim , ncol = 4)
#container matrix for the number of non-zero beta-coeff estimated
lasso.nb <- matrix(NA,nrow = n.sim , ncol = 4)
ridge.nb <- matrix(NA,nrow = n.sim , ncol = 4)
elnet.nb <- matrix(NA,nrow = n.sim , ncol = 4)
# 100 lambda values for ridge and lasso
grid <- 10^seq(2,-2,length=100)
# set up (alpha,lambda)grid to search for pair that minimizes CV error
alp.grid <-seq(0.05,0.9,length=10) ; lam.grid <- 10^seq(2,-2,length=20)
set.seed(123)
#### we loop over simulations and record the ME and AUC value each time #####

###############################################################################
############################### EXAMPLE 1 #####################################
for(i in 1:n.sim) {
        # variance-covariance matrix fill with correlation
        Sigma <- matrix(NA,n.cov,n.cov)
        for(j in 1:n.cov){
                for(k in 1:n.cov) Sigma[j,k] <- 0.5^abs(j-k)
```

```
        }
        diag(Sigma) <-1  # set diagonal to 1
        # N (200) random draws of 1000 covariates with mean 0 and variance Sigma
        X <- MASS::mvrnorm(N, rep(0,n.cov),Sigma)
        dim(X) # p >> n
        # beta-coefficients
        beta <- c(runif(122,2,5),rep(0,878))
        # response variable Y simulation
        y <- apply(X,MARGIN = 1,FUN = function(x) rbinom(1,1,1/(1+exp(-t(x)%*%beta))))
        # split into training and test data
        train <- sample(c(1:N),size = 120)

        ################### Ridge
        # 10-fold cross-validation on ridge to find best of 100 lambda value
        cv.ridge <-cv.glmnet(X[train,],y[train],alpha=0,lambda=grid,
        nfolds = 10,family="binomial")
        ridge.model <- glmnet(X[train,],y[train],alpha=0,
        lambda=cv.ridge$lambda.min,family="binomial")
        # predict outcome using the model with the best lambda
        r.pred.prob <- predict(ridge.model,newx=X[-train,],type = "response" )
        r.pred.classes <- ifelse(r.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        #Misclassification error rate (ME)
        obs.classes <- y[-train]
        ridge.me[i,1] <- mean(r.pred.classes != obs.classes)
        #AUC value
        ridge.auc[i,1] <-auc(y[-train],r.pred.prob )
        # number of non-zero beta-coefficients for ridge
        ridge.nb[i,1]<- length(as.matrix(coef(ridge.model))[which(coef(ridge.model)[-1]!=0),1])

        ######################Lasso
        # 10-fold cross-validation on Lasso to find best of 100 lambda values
        cv.lasso <-cv.glmnet(X[train,],y[train],alpha=1,lambda=grid,
        nfolds = 10,family="binomial")
        lasso.model <- glmnet(X[train,],y[train],alpha=1,
        lambda=cv.lasso$lambda.min,family="binomial")
        # predict outcome using the model with the best lambda
        l.pred.prob <- predict(lasso.model,newx=X[-train,],type = "response")
        l.pred.classes <- ifelse(l.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
        obs.classes <- y[-train]
        lasso.me[i,1]<- mean(l.pred.classes!= obs.classes)
        #AUC value
        lasso.auc[i,1]<- auc(y[-train],l.pred.prob )
        # number of non-zero  beta-coefficients for Lasso regression
        lasso.nb[i,1]<- length(as.matrix(coef(lasso.model))[which(coef(lasso.model)[-1]!=0),1])

        ###################### Elastic-Net
        y1<-as.factor(y)
        data<-as.data.frame(cbind(y1,X))
        test.data <- data[-train,]
        train.data<-data[train,]
        # set up cross validation method for train function
        control<-trainControl(method = "cv",number = 10)
        #set up search grid for alpha and lambda parameters
        srchgrid<-expand.grid(alpha=alp.grid,lambda=lam.grid)
```

```
93        #Training Elastic Net regression:perform CV forecasting y level based on all features
94        cv.elnet<-train(y1~.,data=train.data,method="glmnet",trControl=control,
95        tuneGrid=srchgrid)
96        # Elastic net regression  model
97        op.alp<-cv.elnet$bestTune$alpha
98        op.lam<-cv.elnet$bestTune$lambda
99        elnet.model<-glmnet(X[train,],y[train],alpha=op.alp,lambda=op.lam,family="binomial")
100       # predict outcome using the model
101       eln.pred.prob<-predict(elnet.model,newx=X[-train,],type = "response")
102       eln.pred.classes<- ifelse(eln.pred.prob > 0.5, 1,0)
103       ## Model accuracy :
104       # Misclassification error rate (ME)
105       elnet.me[i,1]<- mean(eln.pred.classes!=obs.classes)
106       # AUC value
107       elnet.auc[i,1]<-auc(y[-train],eln.pred.prob)
108       # number of non-zero beta coefficients for Elastic-net
109       elnet.nb[i,1]<- length(as.matrix(coef(elnet.model))[which(coef(elnet.model)[-1]!=0),1])
110}
111################################################################################
112################################ EXAMPLE 2 ######################################
113for(i in 1:n.sim) {
114       # variance-covariance matrix fill with correlation
115       Sigma <- matrix(NA,n.cov,n.cov)
116       for(j in 1:n.cov){
117              for(k in 1:n.cov) Sigma[j,k] <- 0.5^abs(j-k)
118       }
119       diag(Sigma) <-1  # set diagonal to 1
120       # N (200) random draws of 1000 covariates with mean 0 and variance Sigma
121       X <- MASS::mvrnorm(N,rep(0,n.cov),Sigma)
122       dim(X) # p >> n
123       # beta-coefficients
124       beta <- c(rep(0.8,n.cov))
125       # response variable Y simulation
126       y <- apply(X,MARGIN = 1,FUN = function(x) rbinom(1,1,1/(1+exp(-t(x)%*%beta))))
127       # slit into training and test data
128       train <- sample(c(1:N),size = 120)
129
130       ################### Ridge
131       # 10-fold cross-validation on ridge to find best of 100 lambda value
132       cv.ridge <-cv.glmnet(X[train,],y[train],alpha=0,lambda=grid,
133       nfolds = 10,family="binomial")
134       ridge.model <- glmnet(X[train,],y[train],alpha=0,
135       lambda=cv.ridge$lambda.min,family="binomial")
136       # predict outcome using the model with the best lambda
137       r.pred.prob <- predict(ridge.model,newx=X[-train,],type = "response" )
138       r.pred.classes <- ifelse(r.pred.prob > 0.5, 1,0)
139       ## Model accuracy :
140       #Misclassification error rate (ME)
141       obs.classes <- y[-train]
142       ridge.me[i,2] <- mean(r.pred.classes != obs.classes)
143       #AUC value
144       ridge.auc[i,2] <-auc(y[-train],r.pred.prob )
145       # number of non-zero beta-coefficients for ridge
146       ridge.nb[i,2]<- length(as.matrix(coef(ridge.model))[which(coef(ridge.model)[-1]!=0),1])
147
148       ######################Lasso
149       # 10-fold cross-validation on Lasso to find best of 100 lambda values
```

```
        cv.lasso <-cv.glmnet(X[train,],y[train],alpha=1,lambda=grid,
        nfolds = 10,family="binomial")
        lasso.model <- glmnet(X[train,],y[train],alpha=1,
        lambda=cv.lasso$lambda.min,family="binomial")
        # predict outcome using the model with the best lambda
        l.pred.prob <- predict(lasso.model,newx=X[-train,],type = "response")
        l.pred.classes <- ifelse(l.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
        obs.classes <- y[-train]
        lasso.me[i,2]<- mean(l.pred.classes!= obs.classes)
        #AUC value
        lasso.auc[i,2]<- auc(y[-train],l.pred.prob )
        #number  of non-zero beta-coefficients for lasso
        lasso.nb[i,2]<- length(as.matrix(coef(lasso.model))[which(coef(lasso.model)[-1]!=0),1])
        ######################## Elastic-Net
        y1<-as.factor(y)
        data<-as.data.frame(cbind(y1,X))
        test.data <- data[-train,]
        train.data<-data[train,]
        # set up cross validation method for train function
        control<-trainControl(method = "cv",number = 10)
        #set up search grid for alpha and lambda parameters
        srchgrid<-expand.grid(alpha=alp.grid,lambda=lam.grid)
        #Training Elastic Net regression:perform CV forecasting y level based on all features
        cv.elnet<-train(y1~.,data=train.data,method="glmnet",trControl=control,
        tuneGrid=srchgrid)
        # Elastic net regression  model
        op.alp<-cv.elnet$bestTune$alpha
        op.lam<-cv.elnet$bestTune$lambda
        elnet.model<-glmnet(X[train,],y[train],alpha=op.alp,lambda=op.lam,family="binomial")
        # predict outcome using the model
        eln.pred.prob<-predict(elnet.model,newx=X[-train,],type = "response")
        eln.pred.classes<- ifelse(eln.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
        elnet.me[i,2]<- mean(eln.pred.classes!=obs.classes)
        # AUC value
        elnet.auc[i,2]<-auc(y[-train],eln.pred.prob)
        #number of non-zero beta coefficients for Elastic-net
        elnet.nb[i,2]<- length(as.matrix(coef(elnet.model))[which(coef(elnet.model)[-1]!=0),1])
}
##############################################################################
############################### EXAMPLE 3 ####################################
for(i in 1:n.sim) {
        # variance-covariance matrix fill with correlation
        Sigma <- matrix(NA,n.cov,n.cov)
        for(j in 1:n.cov){
                for(k in 1:n.cov) Sigma[j,k] <- 0.9^abs(j-k)
        }
        diag(Sigma) <-1  # set diagonal to 1
        # N (200) random draws of 1000 covariates with mean 0 and variance Sigma
        X <- MASS::mvrnorm(N,rep(0,n.cov),Sigma)
        dim(X) # p >> n
        # beta-coefficients
        beta <-rep(c(rep(2,125),rep(0,125)),4)
        # response variable Y simulation
```

```
207         y <- apply(X,MARGIN = 1,FUN = function(x) rbinom(1,1,1/(1+exp(-t(x)%*%beta))))
208         # slit into training and test data
209         train <- sample(c(1:N),size = 120)
210 
211         ################### Ridge
212         # 10-fold cross-validation on ridge to find best of 100 lambda value
213         cv.ridge <-cv.glmnet(X[train,],y[train],alpha=0,lambda=grid,
214         nfolds = 10,family="binomial")
215         ridge.model <- glmnet(X[train,],y[train],alpha=0,
216         lambda=cv.ridge$lambda.min,family="binomial")
217         # predict outcome using the model with the best lambda
218         r.pred.prob <- predict(ridge.model,newx=X[-train,],type = "response" )
219         r.pred.classes <- ifelse(r.pred.prob > 0.5, 1,0)
220         ## Model accuracy :
221         #Misclassification error rate (ME)
222         obs.classes <- y[-train]
223         ridge.me[i,3] <- mean(r.pred.classes != obs.classes)
224         #AUC value
225         ridge.auc[i,3] <-auc(y[-train],r.pred.prob )
226         # number of non-zero beta-coefficients for ridge
227         ridge.nb[i,3]<- length(as.matrix(coef(ridge.model))[which(coef(ridge.model)[-1]!=0),1])
228         ######################Lasso
229         # 10-fold cross-validation on Lasso to find best of 100 lambda values
230         cv.lasso <-cv.glmnet(X[train,],y[train],alpha=1,lambda=grid,
231         nfolds = 10,family="binomial")
232         lasso.model <- glmnet(X[train,],y[train],alpha=1,
233         lambda=cv.lasso$lambda.min,family="binomial")
234         # predict outcome using the model with the best lambda
235         l.pred.prob <- predict(lasso.model,newx=X[-train,],type = "response")
236         l.pred.classes <- ifelse(l.pred.prob > 0.5, 1,0)
237         ## Model accuracy :
238         # Misclassification error rate (ME)
239         obs.classes <- y[-train]
240         lasso.me[i,3]<- mean(l.pred.classes!= obs.classes)
241         #AUC value
242         lasso.auc[i,3]<- auc(y[-train],l.pred.prob )
243         #number  of non-zero beta-coefficients for lasso
244         lasso.nb[i,3]<- length(as.matrix(coef(lasso.model))[which(coef(lasso.model)[-1]!=0),1])
245 
246         ######################## Elastic-Net
247         y1<-as.factor(y)
248         data<-as.data.frame(cbind(y1,X))
249         test.data <- data[-train,]
250         train.data<-data[train,]
251         # set up cross validation method for train function
252         control<-trainControl(method = "cv",number = 10)
253         #set up search grid for alpha and lambda parameters
254         srchgrid<-expand.grid(alpha=alp.grid,lambda=lam.grid)
255         #Training Elastic Net regression:perform CV forecasting y level based on all features
256         cv.elnet<-train(y1~.,data=train.data,method="glmnet",trControl=control,
257         tuneGrid=srchgrid)
258         # Elastic net regression  model
259         op.alp<-cv.elnet$bestTune$alpha
260         op.lam<-cv.elnet$bestTune$lambda
261         elnet.model<-glmnet(X[train,],y[train],alpha=op.alp,lambda=op.lam,family="binomial")
262         # predict outcome using the model
263         eln.pred.prob<-predict(elnet.model,newx=X[-train,],type = "response")
```

```
        eln.pred.classes<- ifelse(eln.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
        elnet.me[i,3]<- mean(eln.pred.classes!=obs.classes)
        # AUC value
        elnet.auc[i,3]<-auc(y[-train],eln.pred.prob)
        #number of non-zero beta coefficients for Elastic-net
        elnet.nb[i,3]<- length(as.matrix(coef(elnet.model))[which(coef(elnet.model)[-1]!=0),1])
}
#############################################################################
############################### EXAMPLE 4 ###################################
for(i in 1:n.sim) {
        # variance-covariance matrix fill with correlation
        Sigma <- matrix(0,n.cov,n.cov)
        for(j in 1:n.cov/2){
                for(k in 1:n.cov/2) Sigma[j,k] <- 0.5^abs(j-k)
        }
        diag(Sigma) <-1  # set diagonal to 1
        # N (200) random draws of 1000 covariates with mean 0 and variance Sigma
        X <- MASS::mvrnorm(N,rep(0,n.cov),Sigma)
        dim(X) # p >> n
        # beta-coefficients
        beta <-c(rep(3,500),rep(0,500))
        # response variable Y simulation
        y <- apply(X,MARGIN = 1,FUN = function(x) rbinom(1,1,1/(1+exp(-t(x)%*%beta))))
        # slit into training and test data
        train <- sample(c(1:N),size = 120)
        ################### Ridge

        # 10-fold cross-validation on ridge to find best of 100 lambda value
        cv.ridge <-cv.glmnet(X[train,],y[train],alpha=0,lambda=grid,
        nfolds = 10,family="binomial")
        ridge.model <- glmnet(X[train,],y[train],alpha=0,
        lambda=cv.ridge$lambda.min,family="binomial")
        # predict outcome using the model with the best lambda
        r.pred.prob <- predict(ridge.model,newx=X[-train,],type = "response" )
        r.pred.classes <- ifelse(r.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        #Misclassification error rate (ME)
        obs.classes <- y[-train]
        ridge.me[i,4] <- mean(r.pred.classes != obs.classes)
        #AUC value
        ridge.auc[i,4] <-auc(y[-train],r.pred.prob )
        # number of non-zero beta-coefficients for ridge
        ridge.nb[i,4]<- length(as.matrix(coef(ridge.model))[which(coef(ridge.model)[-1]!=0),1])

        ######################Lasso
        # 10-fold cross-validation on Lasso to find best of 100 lambda values
        cv.lasso <-cv.glmnet(X[train,],y[train],alpha=1,lambda=grid,
        nfolds = 10,family="binomial")
        lasso.model <- glmnet(X[train,],y[train],alpha=1,
        lambda=cv.lasso$lambda.min,family="binomial")
        # predict outcome using the model with the best lambda
        l.pred.prob <- predict(lasso.model,newx=X[-train,],type = "response")
        l.pred.classes <- ifelse(l.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
```

```
        obs.classes <- y[-train]
        lasso.me[i,4]<- mean(l.pred.classes!= obs.classes)
        #AUC value
        lasso.auc[i,4]<- auc(y[-train],l.pred.prob )
        #number  of non-zero beta-coefficients for lasso
        lasso.nb[i,4]<- length(as.matrix(coef(lasso.model))[which(coef(lasso.model)[-1]!=0),1])

        ######################## Elastic-Net
        y1<-as.factor(y)
        data<-as.data.frame(cbind(y1,X))
        test.data <- data[-train,]
        train.data<-data[train,]
        # set up cross validation method for train function
        control<-trainControl(method = "cv",number = 10)
        #set up search grid for alpha and lambda parameters
        srchgrid<-expand.grid(alpha=alp.grid,lambda=lam.grid)
        #Training Elastic Net regression:perform CV forecasting y level based on all features
        cv.elnet<-train(y1~.,data=train.data,method="glmnet",trControl=control,
        tuneGrid=srchgrid)
        # Elastic net regression  model
        op.alp<-cv.elnet$bestTune$alpha
        op.lam<-cv.elnet$bestTune$lambda
        elnet.model<-glmnet(X[train,],y[train],alpha=op.alp,lambda=op.lam,family="binomial")
        # predict outcome using the model
        eln.pred.prob<-predict(elnet.model,newx=X[-train,],type = "response")
        eln.pred.classes<- ifelse(eln.pred.prob > 0.5, 1,0)
        ## Model accuracy :
        # Misclassification error rate (ME)
        elnet.me[i,4]<- mean(eln.pred.classes!=obs.classes)
        # AUC value
        elnet.auc[i,4]<-auc(y[-train],eln.pred.prob)
        #number of non-zero beta coefficients for Elastic-net
        elnet.nb[i,4]<- length(as.matrix(coef(elnet.model))[which(coef(elnet.model)[-1]!=0),1])
}

#############################################################################
######## result Matrix of ME and AUC values for each simulations ##########

## we take column mean to get the Average ME over n.sim simulations and
#create an outcome object where the rows contain ridge,lasso,el-net
#and full logistic results respectively
Ave.ME.results<-rbind(apply(ridge.me ,2,mean),apply(lasso.me ,2,mean),
apply(elnet.me ,2,mean), apply(full.model.me ,2,mean)  )
rownames(Ave.ME.results)<-c("Ridge","Lasso","Elastic␣Net","␣full␣logistic␣model")
colnames(Ave.ME.results)<-c("ME.ave_Exp1","ME.ave_Exp2","ME.ave_Exp3","ME.ave_Exp4")
### Now proceed the same way and store Standard deviation ME over n.sim
Sd.ME.results<-rbind(apply(ridge.me ,2,sd),apply(lasso.me ,2,sd),
apply(elnet.me ,2,sd), apply(full.model.me ,2,sd))
rownames(Sd.ME.results)<-c("Ridge","Lasso","Elastic␣Net","␣full␣logistic␣model")
colnames(Sd.ME.results)<-c("ME.sd_Exp1","ME.sd_Exp2","ME.sd_Exp3","ME.sd_Exp4")
### outcome object containing results Average AUC over n.sim simulations
Ave.AUC.results<-rbind(apply(ridge.auc ,2,mean),apply(lasso.auc ,2,mean),
apply(elnet.auc ,2,mean), apply(full.model.auc ,2,mean)  )
rownames(Ave.AUC.results)<-c("Ridge","Lasso","Elastic␣Net","␣full␣logistic␣model")
colnames(Ave.AUC.results)<-c("AUC.ave_Exp1","AUC.ave_Exp2","AUC.ave_Exp3","AUC.ave_Exp4")
### outcome object containing results standard deviation AUC over n.sim simulations
Sd.AUC.results<-rbind(apply(ridge.auc ,2,sd),apply(lasso.auc ,2,sd),
```

```
apply(elnet.auc ,2,sd), apply(full.model.auc ,2,sd))
rownames(Sd.AUC.results)<-c("Ridge","Lasso","Elastic Net","full logistic model")
colnames(Sd.AUC.results)<-c("AUC.sd_Exp1","AUC.sd_Exp2","AUC.sd_Exp3","AUC.sd_Exp4")
### Outcome object containing results Average of non-zero beta coefficients estimated
# over n.sim simulations
Ave.nbc.results<-rbind(apply(ridge.nb ,2,mean),apply(lasso.nb ,2,mean),
apply(elnet.nb,2,mean), apply(full.model.nb,2,mean) )
rownames(Ave.nbc.results)<-c("Ridge","Lasso","Elastic Net","full logistic model")
colnames(Ave.nbc.results)<-c("nbc.ave_Exp1","nbc.ave_Exp2","nbc.ave_Exp3","nbc.ave_Exp4")

############################## RESUME ##########################
Simulation.Results<-list(Ave.ME=Ave.ME.results ,Sd.ME=Sd.ME.results ,
Ave.AUC=Ave.AUC.results ,Sd.AUC= Sd.AUC.results ,Ave.Num.coef=Ave.nbc.results)
Simulation.Results
```

A.2 R codes for real data example

```
1  ########################## REAL DATA EXAMPLE ##########################
2  #Loading required library
3  library(minfi);library(here);library(readr);library(SummarizedExperiment)
4  library(caret);library(glmnet)
5  ################## let's read in the data
6  DNA.methylation.data<-readRDS(here("datasets/methylation.rds"))
7  DNA.methylation.data # we see that this object has "dim()=5000*37" p>>n
8  # Extract the matrix of methylation M-values
9  methyl.matrix<-assay(DNA.methylation.data)
10 # transpose to have features as column and samples as rows
11 methyl.matrix<-t(assay(DNA.methylation.data))
12 # view dimension of methylation matrix
13 dim(methyl.matrix)
14 # examine the metadata,phenotypes and grouping relating to this data
15 head(pData(DNA.methylation.data)) # for the first 6 samples
16 ######### We will focus on the association between age and methylation ########
17 Age<-DNA.methylation.data$Age
18 #### let us check out what happens if we try fit a linear model to the data ###
19 # R will run a multivariate regression model in which each of the column in
20 # methyl.matrix is used as predictor.
21 linear_model_fit <- lm(Age~methyl.matrix)
22 summary(linear_model_fit)
23 # singularities
24 XtX <- t(methyl.matrix)%*%methyl.matrix
25 det(XtX)  # we can't fit standard linear model to this high-dimensional data.
26 ###### Now we'll work with set of features known to be associated with Age from
27 #                        a paper by "Horvath et al."                          #
28 # read in the data
29 coefhorvath<-readRDS(here("datasets/coefHorvath.rds"))
30 dim(coefhorvath); class(coefhorvath)
31 features<-coefhorvath[1:20,]$CpGmarker
32 horv.matrix<-methyl.matrix[,features]
33 dim(horv.matrix)   # not technically high-dimensional data
34 # Generate an index to split the data into train and test set
35 set.seed(50)
36 train<-sample(nrow(methyl.matrix),27)
37 train.horv.matrix<-horv.matrix[train,] ; train.Age<-Age[train]
```

```
test.horv.matrix<-horv.matrix[-train,] ; test.Age<-Age[-train]
#############################################################################
########### OLS regression Vs Ridge regression on "horv.matrix" data ########
## investigate correlations
corr.matrix<-cor(train.horv.matrix)
write.csv2(corr.matrix,file="corr.matrix.horv.csv")
# using heat map
heatmap(train.horv.matrix,scale = "column")
## multilinear regression fit
horv.lm.fit<-lm(train.Age~.,data = as.data.frame(train.horv.matrix))
summary(horv.lm.fit)
# Check mean squared error on the model.
horv.lm.mse<-mean(residuals(horv.lm.fit)^2)
horv.lm.mse
# examine the MSE on the test Data
pred.lm<-predict(horv.lm.fit,newdata = as.data.frame(test.horv.matrix))
MSE.lm<-mean((test.Age-pred.lm)^2)
MSE.lm
## Ridge regression fit
# 100 lambda values for ridge and lasso
grid <- 10^seq(2,-2,length=100)
# performing Leave One Out CV to search for the best lambda
cv.ridge<- cv.glmnet(x=train.horv.matrix,y=train.Age,nfolds =27,alpha=0,lambda =grid  )
ridge.fit<-glmnet(x=train.horv.matrix,y=train.Age,lambda = cv.ridge$lambda.min,alpha=0)
# plot of test MSE's vs lambda values
#plot showing how estimated coefficients change as we increase the penalty, "lambda"
ridg.fit<-glmnet(x=train.horv.matrix,y=train.Age,alpha=0)
dev.new() ;plot(cv.ridge);dev.new() ;plot(ridg.fit,xvar="lambda")
abline(v=log(cv.ridge$lambda.min),lty="dashed");abline(h=0,lty="dashed")
# examine MSE on test data
pred.ridge<-predict(ridge.fit,newx=test.horv.matrix)
MSE.ridge<-mean((test.Age-pred.ridge)^2);MSE.ridge
####### Which performs better, Ridge or OLS ?
min(c(MSE.ridge,MSE.lm))
# plot predicted Ages for both method against the true Ages
lim<-range(c(pred.lm,test.Age,pred.ridge));dev.new();par(mfrow=1:2)
plot(test.Age,pred.lm,xlim=lim,ylim=lim,pch=19) ;abline(0:1,lty="dashed")
plot(test.Age,pred.ridge,xlim=lim,ylim=lim,pch=19) ;abline(0:1,lty="dashed")
## display coefficients estimated
horv.coefs<-cbind(coef(horv.lm.fit),coef(ridge.fit))
colnames(horv.coefs)<-c("lm_coefs","ridge_coefs");horv.coefs
#############################################################################
###########   LASSO regression on DNA.methylation.data ####################

## examine correlations using the pearson heatmap
heatmap(methyl.matrix,scale = "column",col=cm.colors(256))
# perform 10-folds CV to find the best lambda value
cv.lasso<-cv.glmnet(methyl.matrix,Age,alpha=1,lambda = grid ,nfolds=10)
lasso.fit<-glmnet(methyl.matrix,Age,alpha=1,lambda = cv.lasso$lambda.min)
# plot of test MSE's vs lambda values
#plot showing how estimated coefficients change as we increase the penalty, "lambda"
lass.fit<-glmnet(methyl.matrix,Age,alpha=1);dev.new() ;plot(cv.lasso)
dev.new() ;plot(lass.fit,xvar="lambda");abline(v=log(cv.lasso$lambda.min),lty="dashed")
# view coefficients of the model
lasso_coefficients <- coef(lasso.fit);lasso_coefficients
# view selected variables performed by Lasso regression
selected_coefs <- as.matrix(lasso_coefficients)[which(lasso_coefficients !=0),1]
```

```
selected_features<-names(selected_coefs);selected_features;length(selected_features)
## compare features selected with Horvath signature
intersect(selected_features,coefhorvath$CpGmarker) # we selected some of the same feature
length(intersect(selected_features,coefhorvath$CpGmarker))
## Lasso Vs Ridge coefficients paths
dev.new() ;par(mfrow=c(2,1));plot(ridg.fit,xvar="lambda",main="ridge case")
plot(lass.fit,xvar="lambda",main="lasso case")
##############################################################################
################ Ridge regression on DNA.methylation.data ###################

# perform 10-folds CV to find the best lambda value
cv.r<-cv.glmnet(methyl.matrix,Age,alpha=0,lambda = grid ,nfolds=10)
Ridge.fit<-glmnet(methyl.matrix,Age,alpha=0,lambda = cv.r$lambda.min)
# plot of test MSE's vs lambda values
#plot showing how estimated coefficients change as we increase the penalty, "lambda"
Ridg.fit<-glmnet(methyl.matrix,Age,alpha=0);dev.new() ;plot(cv.r);dev.new()
plot(Ridg.fit,xvar="lambda");abline(v=log(cv.r$lambda.min),lty="dashed")
# view coefficients of the model
ridge_coefficients <- coef(Ridge.fit);ridge_coefficients
##############################################################################
######### Blending Ridge regression and the LASSO : Elastic nets #############

# set up (alpha,lambda)grid to search for pair that minimizes CV error
# using "caret package
alp.grid <-seq(0.05,0.9,length=10) ; lam.grid <- 10^seq(2,-2,length=20)
data<-as.data.frame(cbind(Age=Age,methyl.matrix))
# set up cross validation method for train function
control<-trainControl(method = "cv",number = 10)
#set up search grid for alpha and lambda parameters
srchgrid<-expand.grid(alpha=alp.grid,lambda=lam.grid)
#Training Elastic Net regression:perform CV forecasting y level based on all features
cv.elnet<-train(Age~.,data=data,method="glmnet",trControl=control,tuneGrid=srchgrid)
cv.elnet
# plot CV performance
dev.new();plot(cv.elnet)
# Elastic net regression  model
op.alp<-cv.elnet$bestTune$alpha; op.lam<-cv.elnet$bestTune$lambda
elnet.model<-glmnet(methyl.matrix,Age,alpha=op.alp,lambda=op.lam)
### Lasso Vs Elastic coefficients paths (setting "alpha=op.alp")
eln.model<-glmnet(methyl.matrix,Age,alpha=op.alp);dev.new()
par(mfrow=c(2,1));plot(eln.model,main="Elastic Net case"); plot(lass.fit,main="lasso case")
### compare the coefficients with the LASSO model
elnet_coefs<-coef(elnet.model)
sum(elnet_coefs[,1]==0)     #number of coefficients set to zero for "elnet"
sum(lasso_coefficients[,1]==0) # number of coefficients set to zero for LASSO
# plot Lasso coefficients against Elastic Net coefficients
dev.new()
plot(lasso_coefficients[,1],elnet_coefs[,1],pch=19,xlab="Lasso coefficients",
ylab="Elastic net coefficients"); abline(0:1,lty="dashed",col="blue")
# compare features remaining in the model with Horvath signature
elnet.rm.features<-names(as.matrix(elnet_coefs)[which(elnet_coefs!=0),1])
elnet.rm.features; length(elnet.rm.features)
intersect(elnet.rm.features,coefhorvath$CpGmarker)
length(intersect(elnet.rm.features,coefhorvath$CpGmarker))
# export the estimated coeffitients provide by the tree method
result_coefs<-cbind(lasso_coefficients,ridge_coefficients,elnet_coefs)
colnames(result_coefs)<-c("lasso_coefs","ridge_coefs","eln_coefs")
```

```
result_coefs
write.csv2(result_coefs, file = "coefs_DNA.methyl.data.csv")
```

A.3 R code for ridge regression over measurement error ridden data

```
# loading require package.
library(glmnet)
n=500 ; p=200 #number of samples and number of covariates
n.sim=100 # number of Monte Carlo simulation
grid <- 10^seq(2,-2,length=100) # 100 lambda values for Ridge
### container matrix ###
# container matrix for mean square error(MSE) and prediction error for
# "true linear model (lm), naive linear model, corrected lm, true ridge
#, naive ridge and corrected ridge regression.
lm.res= matrix(NA,nrow = n.sim , ncol = 3)
ridge.res=matrix(NA,nrow = n.sim , ncol = 3)
colnames(lm.res)<-c("t.lm.mse","n.lm.mse","cor.lm.mse")
colnames(ridge.res)<-c("t.ridge.mse","n.ridge.mse","cor.ridge.mse")
lm.res2= matrix(NA,nrow = n.sim , ncol = 3)
ridge.res2=matrix(NA,nrow = n.sim , ncol = 3)
colnames(lm.res2)<-c("t.lm.pe","n.lm.pe","cor.lm.pe")
colnames(ridge.res2)<-c("t.ridge.pe","n.ridge.pe","cor.ridge.pe")
######### Loop over simulation and record MSE and PE value each time #########
for(i in 1:n.sim){
        ###### model setup
        # variance-covariance matrix of true unobserved values with high correlation
        Sigma_X <- matrix(NA,p,p)
        for(j in 1:p){
                for(k in 1:p) Sigma_X[j,k] <- 0.9^abs(j-k)}
        X <- MASS::mvrnorm(n,rep(0,p),Sigma_X) #true variables
        Sigma_U<-diag(x=0.75 ,nrow = p, ncol = p) #measurement error covariance matrix (assume here to be kn
        U<- MASS::mvrnorm(n,rep(0,p),Sigma_U)
        W <- X + U  #measurement matrix ( observed values)
        beta<-runif(p,1,4) #coefficient
        y <- X %*% beta + rnorm(n, sd = 1);y<-scale(y);X<-scale(X);W<-scale(W) #Response
        train <- sample(c(1:n),size = 400) # split into training and test data
        ###### fit true Linear model on training data
        t.lm.fit<-lm(y[train]~X[train,])
        hat.beta<-coef(t.lm.fit)[-1] # estimated coefficient
        lm.res[i,1]<-mean((hat.beta-beta)^2) # MSE
        pred.t.lm<-predict(t.lm.fit,newdata = as.data.frame(X[-train,])) #prediction error
        lm.res2[i,1]<-mean((y[-train]-pred.t.lm)^2)
        ###### fit naive linear model
        n.lm.fit<-lm(y[train]~W[train,])
        hat.beta.n<-coef(n.lm.fit)[-1] # estimated coefficient
        lm.res[i,2]<-mean((hat.beta.n-beta)^2) # MSE
        pred.n.lm<-predict(n.lm.fit,newdata = as.data.frame(W[-train,])) # prediction error
        lm.res2[i,2]<-mean((y[-train]-pred.n.lm)^2)
        ###### correct for measurement error in the model
        n1<-n-100 #  reliability matrix "K" estimate
        hat.K<-solve(t(W[train,])%*%W[train,])%*%(t(W[train,])%*%W[train,]-n1*Sigma_U)
        # estimate coefficient under measurement error
```

```
48                hat.beta.me<-solve(hat.K)%*%hat.beta.n
49                lm.res[i,3]<-mean((hat.beta.me-beta)^2) ## MSE
50                lm.res2[i,3]<-mean((y[-train]-W[-train,]%*%hat.beta.me)^2) ## prediction error
51                ####### fit true Ridge regression model
52
53                # 10-folds cross validation to find the optimal "lambda"
54                cv.t.ridge <-cv.glmnet(X[train,],y[train],alpha=0,lambda=grid,nfolds = 10)
55                t.ridge.fit<- glmnet(X[train,],y[train],alpha=0,lambda=cv.t.ridge$lambda.min)
56                hat.beta_R<-coef(t.ridge.fit)[-1]
57                ridge.res[i,1]<-mean((hat.beta_R-beta)^2) ## MSE
58                pred.t.ridge<-pred.ridge<-predict(t.ridge.fit,newx=X[-train,]) # prediction error on test data
59                ridge.res2[i,1]<-mean((y[-train]-pred.t.ridge)^2)
60                ###### fit naive Ridge
61                # 10-folds cross validation to find the optimal "lambda"
62                cv.n.ridge <-cv.glmnet(W[train,],y[train],alpha=0,lambda=grid,nfolds = 10)
63                n.ridge.fit<- glmnet(W[train,],y[train],alpha=0,lambda=cv.n.ridge$lambda.min)
64                hat.beta_nR<-coef(n.ridge.fit)[-1]
65                ridge.res[i,2]<-mean((hat.beta_nR-beta)^2) ## MSE
66                pred.n.ridge<-predict(n.ridge.fit,newx=W[-train,])# prediction error on test data
67                ridge.res2[i,2]<-mean((y[-train]-pred.n.ridge)^2)
68                ##### correct for measurement error in ridge regression
69                # use estimated reliability matrix "hat.K"
70                # perform regular ridge regression of "y" on "W%*%hat.K "
71                # 10-folds cross validation to find the optimal "lambda"
72                cv.cor.ridge <-cv.glmnet(W[train,]%*%hat.K,y[train],alpha=0,lambda=grid,nfolds = 10)
73                cor.ridge.fit<- glmnet(W[train,]%*%hat.K,y[train],alpha=0,lambda=cv.cor.ridge$lambda.min)
74                hat.beta_corR<-coef(cor.ridge.fit)[-1]
75                ridge.res[i,3]<-mean((hat.beta_corR-beta)^2)  ## MSE
76                ridge.res2[i,3]<-mean((y[-train]-W[-train,]%*%hat.beta_corR)^2)  ## prediction error
77        }
78        ######################## Result Matrix of MSE ##############################
79        lm.res ; ridge.res ;lm.res2 ; ridge.res2
80        lm.MSE.res<-rbind(apply(lm.res ,2,mean),apply(lm.res ,2,sd) )
81        R.MSE.res<-rbind(apply(ridge.res ,2,mean),apply(ridge.res ,2,sd) )
82        lm.PE.res<-rbind(apply(lm.res2 ,2,mean),apply(lm.res2 ,2,sd) )
83        R.PE.res<-rbind(apply(ridge.res2 ,2,mean),apply(ridge.res2 ,2,sd) )
84        #outcome object containing result average and standard deviation for each method
85        simulation.res1<-cbind(lm.MSE.res,R.MSE.res);simulation.res2<-cbind(lm.PE.res,R.PE.res)
86        rownames(simulation.res1)<-c("Ave","Sd");rownames(simulation.res2)<-c("Ave","Sd")
87        simulation.res1 ; simulation.res2 #display
```

Bibliography

[1] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. 2009.

[2] P. Bühlmann and S. Van De Geer. Statistics for high-dimensional data: methods, theory and applications. Springer Science & Business Media, 2011.

[3] J. P. Buonaccorsi. Measurement error: models, methods, and applications. CRC press, 2010.

[4] E. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. 2007.

[5] R. J. Carroll, D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu. Measurement error in nonlinear models: a modern perspective. Chapman and Hall/CRC, 2006.

[6] R. J. Carroll, D. Ruppert, L. A. Stefanski, and C. M. Crainiceanu. Measurement error in nonlinear models: a modern perspective. Chapman and Hall/CRC, 2006.

[7] C.-L. Cheng and J. W. Van Ness. Statistical regression with measurement error. (No Title), 1999.

[8] A. Datta and H. Zou. Cocolasso for high-dimensional error-in-variables regression. 2017.

[9] A. Datta and H. Zou. A note on cross-validation for lasso under measurement errors. Technometrics, 62(4):549–556, 2020.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. 2004.

[11] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, 96(456):1348–1360, 2001.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010.

[13] W. A. Fuller. Measurement error models. John Wiley & Sons, 2009.

[14] J. Gareth, W. Daniela, H. Trevor, and T. Robert. An introduction to statistical learning: with applications in R. Spinger, 2013.

[15] L. J. Gleser. The importance of assessing measurement reliability in multivariate regression. Journal of the American Statistical Association, 87(419):696–707, 1992.

[16] T. Hastie, R. Tibshirani, and M. Wainwright. Statistical learning with sparsity: the lasso and generalizations. CRC press, 2015.

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

[18] M. Rosenbaum and A. B. Tsybakov. Sparse recovery under matrix uncertainty. The Annals of Statistics, pages 2620–2651, 2010.

[19] A. M. E. Saleh et al. A ridge regression estimation approach to the measurement error model. Journal of Multivariate Analysis, 123:68–84, 2014.

[20] Ø. Sørensen, A. Frigessi, and M. Thoresen. Measurement error in lasso: Impact and likelihood bias correction. Statistica sinica, pages 809–829, 2015.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

[22] S. A. Van De Geer and P. Bühlmann. On the conditions used to prove oracle results for the lasso. 2009.

[23] X. Yan and X. Su. Linear regression analysis: theory and computing. world scientific, 2009.

[24] P. Zhao and B. Yu. On model selection consistency of lasso. The Journal of Machine Learning Research, 7:2541–2563, 2006.