YILDIZ TECHNICAL UNIVERSITY

1911

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PhD THESIS

July, 2015

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NEW TECHNIQUE FOR HIGH DIMENSIONAL DATA: ROBUST LINEAR REGRESSION USING L_1-PENALIZED MM-ESTIMATION

KAMAL DARWISH

PhD THESIS
DEPARTMENT OF STATISTICS
PROGRAM OF STATISTICS

ADVISER
PROF. DR. ALİ HAKAN BÜYÜKLÜ

İSTANBUL, 2015

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NEW TECHNIQUE FOR HIGH DIMENSIONAL DATA: ROBUST LINEAR REGRESSION USING L_1-PENALIZED MM-ESTIMATION

A thesis submitted by Kamal Darwish in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 07.07.2015 in Department of Statistics.

Thesis Adviser
Prof. Dr. Ali Hakan Büyüklü
Yıldız Technical University

Approved By the Examining Committee
Prof. Dr. Ali Hakan Büyüklü
Yıldız Technical University ______________________

Prof. Dr. Adnan Mazmanoğlu, Member
İstanbul Aydın University ______________________

Prof. Dr. Gülay Kıroğlu, Member
Mimar Sınan University ______________________

Doç. Dr. Fatma Noyan, Member
Yıldız Technical University ______________________

Doç. Dr. Atıf Evren, Member
Yıldız Technical University ______________________

To the soul of my Parents.

ACKNOWLEDGEMENTS

First of all I would like to thank Allah almighty for giving me the power and help to accomplish this research and making this work successful. Without his help nothing was possible.

Then I am grateful to my supervisor Dr Prof. Dr. Ali Hakan Büyüklü for his enormous support, valuable advice, encouragement and professional assistance throughout the work of this research.

Then I would like to thank all my family members specially my wife and my children

who have been constant source of motivation, inspiration and support during my whole educational life and also their guidance at each and every step.

Finally I would like to thank all my friends and well to all good people for their encouragement and help.

July, 2015

Kamal DARWISH

TABLE OF CONTENTS

Page

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

LIST OF SYMBOLS

n	The number of observations (rows) in the dataset
p	The total number of covariates
$\boldsymbol{\varepsilon}$	An additive error vector
$\mathbf{r}$	An observable residuals vector
H	The distribution of the contamination
F	The distribution of the non-contaminated part of the data
ϵ	Relatively small positive quantity
ϵ_n^*	The finite breakdown point
γ^*	The gross error sensitivity
$\psi(.)$	The score function
$\rho(.)$	The objective function
$w(.)$	Weight function
$\mathbf{W}$	An n x n diagonal matrix of weights
w_i	The weight associated with row i
α	The proportion of trimming that is performed
h	Trimming constant
b	A constant that controls the estimates robustness
Φ	The standard normal distribution
V(.,.)	The asymptotic variance functional
$f(.)$	The density function
k_{BI}	Tuning constant for the bisquare estimator
k_{HU}	Tuning constant for Huber estimator
$P_\lambda(.)$	The penalty function
λ	The penalty parameter
$\mathbf{H}$	A linear matrix hat operator
$\mathbf{U}$	Orthogonal matrix of order n × p
$\mathbf{V}$	Orthogonal matrix of order p × p
$\mathbf{D}$	A p × p diagonal matrix with the singular entries
t	A parameter that has a one-to-one correspondence with λ
$\hat{S}(\lambda)$	The selected model, for a given λ
S_0	The active set of variables
φ^2	The so-called compatibility constant or restricted eigenvalue
$\hat{\mathscr{S}}$	All possible Lasso sub-models can be computed
$\mathbf{e}_i$	A p-dimensional unit vector with ith element
$\boldsymbol{t}$	Location vector matrix

$\mathbf{V}^*$	Scatter vector matrix
$\mathrm{D}(.)$	The Mahalanobis distance
$\mathbf{R}_0$	Initial correlation matrix
H	A subsample with $\|H\| = h$
$\hat{\boldsymbol{\mu}}_{raw}$	The residual center estimate of the raw sparse LTS estimator
$\hat{\boldsymbol{\mu}}_{reweighted}$	The residual center estimate of the reweighted sparse LTS estimator
$\hat{\sigma}_{\mathrm{raw}}$	The residual scale estimate associated to the raw sparse LTS estimator
r_c^2	Squared centered residuals
$\hat{\sigma}_{reweighted}$	The residual scale estimate of the reweighted sparse LTS estimator
k_{α_w}	The consistency factor
$\boldsymbol{\Lambda}^{(i)}$	The generalized inverse (pseudo-inverse) matrix
$\boldsymbol{r}_-$	The prediction errors vector

LIST OF ABBREVIATIONS

ACM	Association for Computing Machinery
AE	The asymptotic efficiency
AIC	The Akaike Information Criterion
ARE	The asymptotic relative efficiency
BLUE	The best linear unbiased estimator
BIC	Bayesian Information Criterion
CV	Cross Validation
DAR	Distributed at random.
DCAR	Distributed completely at random.
fML	The functional Magnetic Resonance Imaging
FNR	The false negative rate
FPR	The false positive rate
GCV	Generalized Cross Validation
GEM	General gross-error model
GES	Gross error sensitivity
IRLS	Iteratively reweighted least squares
LAD	Least absolute deviations
LARS	Least angle regression
Lasso	The least absolute shrinkage and selection operator
LAV	Least absolute values
LMS	Least median squares
LOOCV	Leave-one-out cross-validation
LTS	Least trimmed squares
MAD	The Median Absolute Deviation
MED	The Median
MLE	Maximum likelihood estimator
MSE	The mean squared error
OLS	The ordinary least squares
OOP	Object oriented programming
RLARS	Robust Least Angle Regression
RMSPE	The root mean squared prediction error
SD	Standard deviation
SVD	Singular value decomposition
UML	The Unified Modeling Language

LIST OF FIGURES

Page

LIST OF TABLES

ABSTRACT

NEW TECHNIQUE FOR HIGH DIMENSIONAL DATA: ROBUST LINEAR REGRESSION USING L_1-PENALIZED MM-ESTIMATION

Kamal DARWISH

Department of Statistics
PhD. Thesis

Adviser: Prof. Dr. Ali Hakan BÜYÜKLÜ

Large datasets, where the number of predictors *p* is larger than the sample sizes *n*, have become very popular in recent years. These datasets pose great challenges for building a linear good prediction model. In addition, when dataset contains a fraction of outliers and other contaminations, linear regression becomes a difficult problem. Therefore, we need methods that are sparse and robust at the same time. In this thesis, we employed the approach of MM estimation and proposed L_1-Penalized MM-estimation (MM-Lasso) as a new estimation method. Our proposed estimator uses sparse LTS estimator as initial estimator to compute penalized M-estimator getting sparse modeli estimation with high breakdown value and good prediction. We implemented MM-Lasso by using C programming language and calling it from R package.

To evaluate our proposed estimator, we extended the SimFrame R package, which is a general framework for simulation studies in statistics. We generated three data models to represent low, moderate and high dimensional data. We also implemented the function for generating the data for the contamination. Simulation study shows that the MM-lasso estimation has better prediction performance than its competitors in the presence of leverage points.

Key words: MM Estimate, Sparse Model, LTS Estimate, Robust Regression, Simulation

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ÖZET

BÜYÜK BOYUTLU VERILER IÇIN YENI BIR TEKNIK: L_1 –CEZALI DOĞRUSAL ROBUST MM-TAHNINCISI

Kamal DARWISH

İstatistik Bölümü
Doktora Tezi

Tez Danışmanı: Prof. Dr. Ali Hakan BÜYÜKLÜ

Son yıllarda büyük veriler çerçevesinde kullanılan p tahmin edicinin (açıklayıcı değişkenli) n gözlem sayısından daha fazla olma durumunda olan modeller oldukça popüler oldular.Bu veri setleri iyi tahmin edilmiş modeller için iyi birer rekabet ortamı oluşturmaktadırlar. Bununla birlikte, veri setlerinde belirli miktarda sapan değerlerin mevcudiyeti ve dahi bazı veri setini bozucu (kontaminasyonlar) unsurların varlığı doğrusal lineer modellerin çözümünü zorlaştırmaktadırlar. Bu durumlarda model çözümleri için metodların seyrek ve robust (dayanıklı) olması istenir. Bu tezde, yeni bir tahmin metodu olarak MM tahmincisi ve L_1- Penalized MM tahmincisi(MM-Lasso) kullanıldı. İleri sürülen tahmin edici, başlangıç tahmin edicisi olarak sparse LTS tahmin edicisi ile M tahmin edicilerini cezalandırarak seyrek model tahminlerini yüksek bozucu değerleri de kapsayarak iyi tahminler vermesi sağlandı.

MM-Lasso C programlama dili ile yazıldı ve R paketi içerisinden de çalıştırılabilir özellik kazandırıldı. İleri sürdüğümüz modeli değerlendirmek için mevcut SimFrame R paketini geliştirdik, bu da istatistiksel olarak simülasyon çalışmaları için bir çerçeve oluşturdu. Üç değişik model geliştirilerek düşük, orta ve büyük boyutlu veriler eldeedildi. Aynı zamanda simülasyon çalışmaları çerçevesinde Kirlenmiş veri oluşturabilmek için fonksiyon geliştirildi. Kaldıraç verilerinin varlığı halinde yapılan incelemelerde MM-Lasso tahmin edicisinin diğer rakiplerinden daha iyi bir performans sergilediği görülmektedir.

Anahtar Kelimeler: MM-Tahmin edicisi, Seyrek Modeli, LTS tahmin edicileri, Robust Regresyon, Simülasyon

CHAPTER 1

INTRODUCTION

1.1 Literature Review

In modern statistical practice in many research areas such as spanning bioinformatics, machine learning, imaging, and signal processing, there is increasing interest in applications containing high-dimensional data sets. These data sets such as ; gene expression microarray data and functional Magnetic Resonance Imaging (fMRI) data, have the total number of variables p is much larger than sample size n, but the number of important variables is typically smaller than n. For example, microarray experiments allow one to measure thousands of variables (genes, proteins) simultaneously. The data sets may contain tens of thousands of predictors, although the effective dimension of the feature space might be much smaller. Therefore, variable selection in regression is important for high-dimensional data analysis to reduce variability and obtain a more interpretable model.

However, in high-dimensional model, the predictors may be highly correlated i.e. multicollinearlity problem occurs which will lead to ill-conditioning in design matrix. Consequences, using the ordinary least squares method (OLS), will lead to misleading results, for example, some of the regression coefficients may be statistically insignificant or have the wrong sign, and they may result in ill-posed statistical inference. Furthermore, when the assumption of normality is violated such as in the case of heavy-tailed errors, the ordinary least squares method (OLS) is not appropriate.

To overcome the multicollinearity problem and reduce the variability, several regularized regression methods have been proposed for fitting multiple regression models in a high dimensional data. One approach of regularized regression can be done by adding a penalty to the objective function on the regression coefficients. This approach shrinks the coefficients and reduces variance at the price of increased bias. Ridge regression, proposed by [1] is L_2-penalized regression method that gives alternative solution to ordinary least squares estimates. Although ridge regression can produce accurate estimates under a large number of predictors, it cannot perform variable selection simultaneously.

In regression, common data-based variable selection procedures, such as forward, backward, stepwise selection and all subset regression, are not suitable in a high dimensional data. Fortunately, for many of applications in high-dimensional setting, there is a prior knowledge of the existence of a sparse representation of the data, either of a hypothesized form or to be discovered by exploration. Sparse linear regression usually describe feature and variable selection problems in high-dimensional linear models. The least absolute shrinkage and selection operator (Lasso) [2] is the popular regularized method in sparsity linear regression models. It has also attractive computational and asymptotic properties (e.g. [3, 4, 5, 6, 7, 8, 9]). Following from the seminal paper of Tibshirani [1], several modifications and extensions of lasso have been developed (e.g. [14, 15, 16]).

Lasso minimizes the residual sum of squares subject to an L_1-norm constraint. Lasso as with ridge regression can shrink the coefficients, but its L_1-penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero, i.e., to produce sparse model estimates that are highly interpretable. Hence, much like best subset selection, the Lasso performs variable selection. It can effectively select important explanatory variables and estimate regression parameters simultaneously. In contrast to classical L_0-penalized variable selection methods, the Akaike Information Criterion (AIC [10]), Bayesian Information Citerion (BIC [11]), Mallows (C_P [12]), and so on, the Lasso is computationally feasible for high-dimensional data. A fast algorithm for computing the Lasso is available through the framework of least angle regression (LARS) [13].

Since Lasso is a penalized least squares method, and ordinary least squares method is extremely sensitive to deviations from the model assumptions, so Lasso is not robust to heavy-tailed error distribution and/or outliers. The breakdown point of the Lasso is 1/n. i.e., only one single outlier can make the Lasso estimate completely unreliable [17].

To produce more robust estimator than Lasso that perform variable selection in high-dimensional data, some methods have been proposed by combining regularized methods with robust regression methods. For example, the least absolute deviations (LAD) regression and Lasso regression are combined to produce an estimator called LAD-lasso [18]. Huber lasso proposed by [19], combines the Huber's criterion loss with a lasso penalty. [20] proposed a weighted version of the LAD-lasso called weighted LAD-lasso that is made resistant to outliers by down-weighting leverage points.

All estimators mentioned until now, are a special case of a more general estimator, the penalized M-estimator [21]. However, using other convex loss functions rather than squared error loss, as done in the Lasso LAD-lasso, has not solved the non robustness of penalized M-estimators with respect to leverage points and result in a breakdown point of $1/n$ [17].

A robust version of ridge regression called Ridge MM-estimator was proposed by [22]. This estimator is an approach of MM estimation that combines high breakdown value estimation with efficient estimation under the normal model. Although Ridge MM-estimator does not produce sparse model estimates, prediction accuracy can be improved by shrinking the coefficients, and the computational issues with high-dimensional robust estimators are overcome due to the regularization.

RLARS proposed by [23], is a robust version of the stepwise algorithm LARS that is computationally very efficient but sensitive to outliers. Although RLARS is robust estimator with respect to leverage points, its algorithm lacks of a natural definition, since this approach does not clearly state which optimization problem is solved.

A popular robust estimator is the least trimmed squares (LTS) estimator [24]. Although its simple definition and fast computation make it interesting for practical application, it cannot be computed for high-dimensional data ($p > n$). Combining the lasso estimator with the LTS estimator developed the sparse LTS-estimator [17]. This estimator can be applied to high-dimensional data with good prediction performance and high robustness. Although, sparse LTS can be applied to high-dimensional data, its efficiency is an issue.

1.2 Objective of the Thesis

The objective of this research is to develop and evaluate regression estimator suitable for high-dimensional data sets, that well performs in the presence of nonnormal errors (outliers) and multicollinearlity problem. This estimator has the following properties:

1. It can produce high-dimensional model fitting with good prediction accuracy and a sparse representation of the predictors in the model.
2. It is not excessively affected by small departures from model assumptions. These departures may include departures from an assumed sample distribution or data departure from the rest of the data (i.e. outliers).
3. It does not cause disaster with larger deviations from the assumed model.

1.3 Hypothesis

In this study, we attempt to combine sparse LTS estimator with penalized M-estimators (Tukey's biweight functions with L_1-penalty). Our combined estimator is an employment of the approach of MM estimation, which was first proposed by [25]. We use the sparse LTS as an initial estimator for computing L_1-penalized M-estimator yields L_1- penalized MM-estimator (MM-Lasso). Thus, MM-Lasso has high breakdown value and efficient estimation. In order to check the performance of our proposed robust method, we run some simulation studies to compare it with others competitor methods.

This thesis is organized as follows. In chapter 2, we present review of robust regression. In chapter 3, we discuss the regularized least squares regression methods. Following the discussion in Chapter 3, chapter 4 discusses robust-regularized regression methods of the related literature and provides a detailed description of our proposed method to compute the robust-sparse regression estimator. Chapter 5 gives some background of simulation, and describes the design, implementation of SimFrame R package and how we can extend the framework. Chapter 5 also presents the results of a simulation study that compares the performance of the estimated models by the root mean squared prediction error (RMSPE). In addition, the concerning sparsity of the estimated models are evaluated by the false positive rate (FPR) and the false negative rate (FNR). Finally, in chapter 6, we conclude the main results obtained of this thesis and discus the future work.

CHAPTER 2

ROBUST REGRESSION

2.1 Matrix notations

In this thesis, we use n to represent the number of distinct data points, or observations and let p to represent the number of predictors. In general, let x_{ij} represent the value of the jth variable for the ith observation, where $i = 1, 2,...,n$ and $j = 1, 2,...,p$., here i will be used to index the samples or observations (from 1 to n) and j will be used to index the variables (from 1 to p). Let **X** denote an $n \times p$ matrix whose (i, j) the element is x_{ij}. That is,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{2.1}$$

Let $\boldsymbol{x}_1, \boldsymbol{x}_2,...,\boldsymbol{x}_n$. denote the rows of **X**. Here $\boldsymbol{x}_i$ is a vector of length p, containing the p variable measurements for the ith observation. That is,

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} \tag{2.2}$$

Let $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p$ denote the columns of **X.** Each is a vector of length n. That is,

$$\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix} \tag{2.3}$$

Then the matrix **X** can be written as

$$\mathbf{X} = (\mathbf{x}_1 \mid . . . \mid \mathbf{x}_p), \tag{2.4}$$

or

$$\mathbf{X} = \begin{pmatrix} \boldsymbol{x}_1' \\ \boldsymbol{x}_2' \\ \vdots \\ \boldsymbol{x}_n' \end{pmatrix} \tag{2.5}$$

where $\boldsymbol{x}_i'$ denotes the transpose of vector $\boldsymbol{x}_i$.

We will use y_i to denote the ith observation of the response. So, we can write the set of all n observations in vector form as

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{2.6}$$

Then the observed data consists of $\{(\boldsymbol{x}_1, y_1),\ (\boldsymbol{x}_2, y_2),\ . . . ,\ (\boldsymbol{x}_n, y_n)\}$, where each $\boldsymbol{x}_i$ is a vector of length p.

2.2 Limitations of ordinary least squares (OLS)

Regression modeling is used to predict one variable from one or more other variables. Normally it involves transforming real data with explanatory variables and a response variable into a linear mathematical equation or expression. The classical linear regression model in matrix form as (e.g. [26], [27])

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \tag{2.7}$$

where $\boldsymbol{y}$ is response vector; $\boldsymbol{\beta} = \left(\beta_1, \ldots, \beta_{\mathrm{p}}\right)'$ is an unknown parameter vector whose values are estimated from the experimental or observational data; $\mathbf{X}$ is design matrix which is based on n iid observations; and $\boldsymbol{\varepsilon} = (\varepsilon_1,...,\varepsilon_n)$ is an additive error vector representing the unexplained random variations in the response. We also consider $\boldsymbol{\varepsilon} \sim N_n\,(\mathbf{0}, \sigma^2\boldsymbol{I})$, this implies that the random vector y has mean $\mathbf{X}\boldsymbol{\beta}$ and variance-covariance matrix $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$.

The ordinary least squares (OLS) is the usual estimation method used to estimate the model coefficients $\boldsymbol{\beta}$. To obtain the least squares estimators for $\boldsymbol{\beta}$ in the linear model, the linear model (2.7) can be rewritten in the form

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{r} \tag{2.8}$$

where $\hat{\boldsymbol{\beta}}$ are estimators of $\boldsymbol{\beta}$ and $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ is an observable residuals vector.

The method of ordinary least squares choose the parameter vector $\boldsymbol{\beta}$ based on the minimization of the residual sum of squares with respect to $\boldsymbol{\beta}$:

$$\underset{\beta \in \Re^{p}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathrm{r}_i^2 = \underset{\beta \in \Re^{p}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i \text{ -} \boldsymbol{x}_i' \hat{\boldsymbol{\beta}})^2 \tag{2.9}$$

By taking the partial derivatives with respect to $\boldsymbol{\beta}$ and setting the equations to zero, these system of equations, also known as the normal equations, are obtained of the form

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y} \tag{2.10}$$

To solve for $\hat{\boldsymbol{\beta}}$, multiply each side of the equation by $(\mathbf{X}'\mathbf{X})^{-1}$ to obtain the ordinary least squares estimator

$$\hat{\beta}_{\mathrm{OLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{2.11}$$

The Gauss–Markov theorem confirms that the ordinary least-squares estimator of the parameter $\boldsymbol{\beta}$ in the classical linear regression model is the best linear unbiased estimators (BLUE). This means that, among the class of linear unbiased estimators for $\boldsymbol{\beta}$, the estimator $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ is the best in the sense that the variances of $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ are minimized.

However, OLS estimator is the best unbiased estimator (BLUE) when error terms are normally distributed. But under conditions of nonnormal distributions, particularly presence of outliers which arise either from heavy tailed distribution or from gross errors can considerably alter the OLS estimation results, and standard errors and confidence intervals are affected badly. In addition, an examination of the residuals may be deceptive because the residuals may seem as a normal distribution even in the presence of gross errors. Therefore, there is a need to modify OLS and obtain robust estimators by down-weight the effect of the outliers.

In regression analysis, the outliers that influence the OLS estimator can be classified in three types as [28]:

- Vertical outlier is that observation ($\boldsymbol{x}_i$,y_i) that deviates from the regression line, but $\boldsymbol{x_i}$ belongs to the majority in x-space. Its presence affects the OLS-estimation and in particular the estimated intercept.
- Good Leverage point is observation ($\boldsymbol{x}_i$,y_i) that deviates from the majority in x-space but follows the regression line. Its presence does not affect the OLS-estimation but it affects statistical inference since OLS estimates can have much larger variances than minimum attainable variances.
- Bad Leverage point is observation ($\boldsymbol{x}_i$,y_i) that deviates from two dimensions (x-sparse and y-sparse) and located far from the true regression line. Its presence affects significantly the OLS-estimation of both the intercept and the slope, OLS estimates will be asymptotically biased even in large sample sizes.

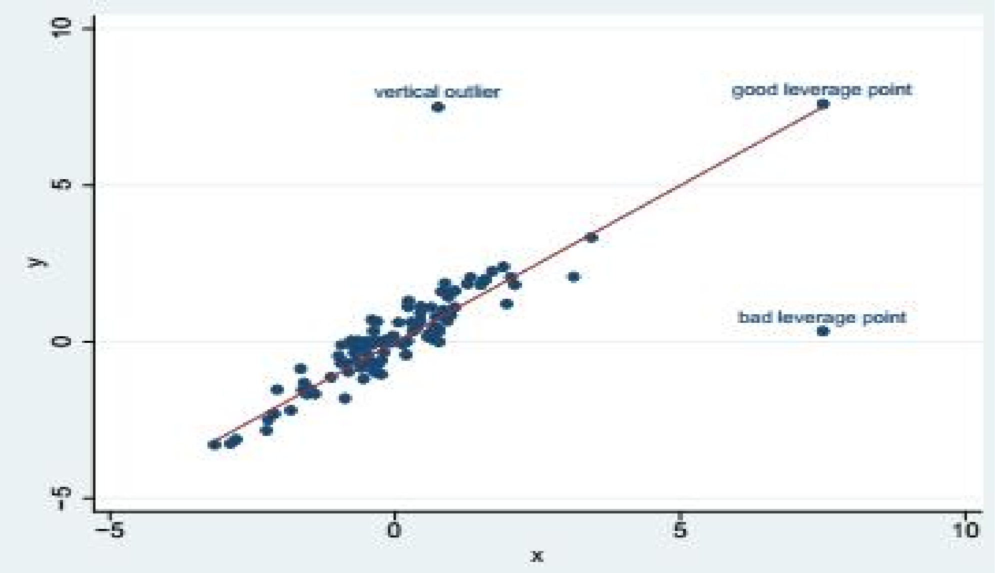

Figure 2.1Non-robustness of OLS in a simple linear regression [29]

2.3 Robustness Concepts

In order to allow for general types of outliers we take the more general gross-error model (GEM) or ϵ-contamination model

$$G(x) = (1 - \epsilon)\, F(x) + \epsilon\, H(x). \qquad (2.12)$$

where H is the distribution of the contamination, relatively small parameter ϵ is the proportion of contamination and F is the distribution of the non-contaminated part of the data. If a normal model is assumed for data, errors can result from the presence of a heaver-tailed distribution i.e. small deviation from the regression model with normal error. Hence, the goal is to achieve robust and relatively small bias estimation of the parameters of F with high efficiency relative to the OLS estimator.

There are three properties that a "good" robust estimator should have [30]:

1. It be should highly efficient when the assumed model is correct (i.e. F is normal).
2. It should only have a minimal effect on performance of estimate with small deviations from the assumed model (i.e. F is in a small neighborhood of normal).
3. It "does not cause catastrophe" with larger deviations from the assumed model.

2.3.1 Some key properties of an estimator

This subsection defines some key properties to determine the strengths and weaknesses of an estimator [24].

Definition 2.1 (Relative efficiency of an estimator) : Let T_1 and T_2 are two estimators, for a population parameter θ, then the efficiency of T_1 relative T_2 is the ratio of its mean squared error T_1 to the mean squared error of T_2 , written

$$\mathrm{RE}\ (T_1,\ T_2) = \frac{\mathrm{MSE}\left[\mathrm{T}_1\right]}{\mathrm{MSE}\left[\mathrm{T}_2\right]}. \tag{2.13}$$

If the two estimators are unbiased, then the efficiency of T_1 relative T_2 is the ratio of variances, written

$$\mathrm{RE}\ (T_1,\ T_2) = \frac{\mathrm{Var}\left[\mathrm{T}_1\right]}{\mathrm{Var}\left[\mathrm{T}_2\right]}. \tag{2.14}$$

The estimator T_1 is more efficient than the estimator T_2 if, for any sample size, $\mathrm{MSE}\left[\mathrm{T}_1\right] \le \mathrm{MSE}\left[\mathrm{T}_2\right]$, which then implies that RE $(T_1,\ T_2) \le 1$.

When the errors are normally distributed the OLS estimators equal to the maximum likelihood estimators (MLE), in this case, the relative efficiency is computed as the mean squared error of the least squares fit divided by the mean squared error of the robust fit. In fact, efficiencies around 90-95% are desirable.

According to [31], in the case of large-sample, relative efficiency is usually calculated in terms of asymptotic efficiency.

Definition 2.2: Let T_{n*} and T_n be asymptotically unbiased for θ. Then, define

1. The **asymptotic relative efficiency** of T_n with respect to T_{n*} at θ is defined as $ARE(T_n, T_{n*}, \theta) = \lim_{n\to\infty} \mathrm{Var}_\theta(T_{n*}) / \mathrm{Var}_\theta(T_n)$, $\theta \in \Theta \subset \Re^p$ be the p-vector of unknown parameters.
2. Let T_{n*}, T_n are asymptotically unbiased for θ, then T_{n*} is called **asymptotically efficient** if $ARE(T_n, T_{n*}, \theta) \leq 1$, $\forall\, \theta \in \Theta$.
3. The **asymptotic efficiency** of T_n is defined as $AE(T_n , \theta) \equiv ARE(T_n, T_{n*} , \theta)$ if T_{n*} is asymptotically efficient.

Definition 2.3 (Regression equivariance): An estimator T is regression equivariant if

$$\mathrm{T}\left(\left\{\left(\boldsymbol{x}_i', \mathrm{y}_i + \boldsymbol{x}_i'\mathbf{v}\right); i = 1,\ldots,n\right\}\right) = \mathrm{T}\left(\left\{\left(\boldsymbol{x}_i', \mathrm{y}_i\right); i = 1,\ldots,n\right\}\right) + \mathbf{v} \tag{2.15}$$

where $\mathbf{v}$ is any column vector.

It means that any additional linear dependence $\boldsymbol{y} \to \boldsymbol{y} + \mathbf{Xv}$ is reflected in the regression vector accordingly $\hat{\boldsymbol{\beta}} \to \hat{\boldsymbol{\beta}} + \mathbf{v}$.

Definition 2.4 (Scale equivariance): An estimator T is said to be scale equivariant if

$$\mathrm{T}\left(\left\{\left(x_i', c\,\mathrm{y}_i\right); i = 1,\ldots,n\right\}\right) \; = c\,\mathrm{T}\left(\left\{\left(x_i', \mathrm{y}_i\right); i = 1,\ldots,n\right\}\right) \tag{2.16}$$

for any constant c. It implies that the fit is essentially independent of the choice of measurement unit for the response variable y. as a result, if $\boldsymbol{y} \to c\,\boldsymbol{y}$ then $\hat{\boldsymbol{\beta}} \to c\hat{\boldsymbol{\beta}}$. So if an estimator is not scale equivariant then the residuals must be standardized before estimating the regression parameter.

Definition 2.5 (Affine equivariance): An estimator T is called affine equivariant if

$$\mathrm{T}\left(\left\{\left(\boldsymbol{x}_i'\mathrm{A}, \mathrm{y}_i\right); i = 1,\ldots,n\right\}\right) \; = \mathrm{A}^{-1}\mathrm{T}\left(\left\{\left(\boldsymbol{x}_i'\mathrm{A}, \mathrm{y}_i\right); i = 1,\ldots,n\right\}\right) \tag{2.17}$$

In other words, affine equivariance means that a linear transformation of the $\boldsymbol{x}_i$ should transform the estimator T accordingly, i.e. $\mathbf{X} \to \mathbf{XA}$, the estimator is transformed accordingly, $\hat{\boldsymbol{\beta}} \to \mathrm{A}^{-1}\hat{\boldsymbol{\beta}}$.

2.3.2 Statistical functional and influence function

Let us define statistical functional which was introduced by von Mises (1936, 1947) and used in the theory of robust estimation.

Definition 2.6 (The statistical functional): Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be a sample from a population with distribution function F and let $T_n = T_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ be a statistic. Then T_n can be written as a functional T of the empirical distribution function F_n, $T_n = T(F_n)$ where T does not depend on n, then we call T a statistical functional.

Definition 2.7 (The linear statistical functional): Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be independent, identically distributed random variables, distributed according to a parametric model $\{F_\theta, \theta \in \Theta \subset \Re^p\}$. Let φ be a real valued function and let

$$T_n = T_n(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n). \tag{2.18}$$

Then the linear statistical functional T (F_θ) is

$$T(F_\theta) = \int \varphi(\boldsymbol{x}) dF_\theta(x) \tag{2.19}$$

Example 2.1: For any distribution F, all its moments are linear functional since T(F) = $\int \boldsymbol{x}^k dF(\boldsymbol{x})$, k =1, . . . , *n*.

We always consider functional that are Fisher consistent. That is T (F) = θ for all θ ∈ Θ. This means that the estimator asymptotically measures the true value.

Definition 2.7 The influence function measures the effect on the estimator T when adding an infinitesimally small amount of contamination at the pointwise z = (x_0, y_0), given a large sample distribution F, and written as

$$\text{IF(z; T, F)} = \lim_{\epsilon \to 0} \frac{T((1-\epsilon)F + \epsilon\delta_z) - T(F)}{\epsilon} = \frac{\partial}{\partial\epsilon}\left[T((1-\epsilon)F + \epsilon\delta_z)\right]|_{\epsilon=0} \tag{2.20}$$

where δz the point mass distribution at z [32, 33, 34]. Note that IF(z; T, F) characterizes the sensitivity of the estimator to slight deviations from the model distribution.

In fact, the influence function consider as a special case of the von Mises derivative of a statistical functional [34, 35]. A functional T is von Mises differentiable at a distribution function F in if there exists a real function IF [. ; T,F] such that

$$\lim_{\epsilon \to 0} \frac{T((1-\epsilon)F + \epsilon H) - T(F)}{\epsilon} = \int \text{IF}(z;\ T,\ F) dH(z). \tag{2.21}$$

By putting H $=\delta_z$, the von Mises derivative becomes the influence function IF[z; T,F]. We noted that if a distribution function H is near F then by applying a von Mises expansion, which resembles a first order Taylor expansion, we get

$$T_n = T(H_n) = T(F) + \int \mathrm{IF}(z;\ T,F)\,d(H-F)(z) + \text{Remainder} \qquad (2.22)$$

The main property of IF is the average influence of all points *z* on the estimation error is zero [36], i.e.

$$E_F\,(IF(z;\ T,F)) = \int \mathrm{IF}(z;\ T,\ F)\,dF(z) = 0, \qquad (2.23)$$

then by using (2.22) we get,

$$T_n = T(F) + \int \mathrm{IF}(z;\ T,F)\,dH(z) + \text{Remainder}.$$

Multiplying by $\frac{1}{\sqrt{n}}$ we get

$$\sqrt{n}\left(T_n - T(F)\right) = \frac{1}{\sqrt{n}} \int \mathrm{IF}(z;\ T,F)\,dH(z) + \text{Remainder} \qquad (2.24)$$

Now, if the "Remainder" is negligible, then by Central limit Theorem, the errors asymptotically becomes a zero mean Gaussian with the variance determined by IF, i.e. $\sqrt{n}\left(T_n - T(F)\right) \xrightarrow{d} N\left(0, AV(T,F)\right)$ where,

$$AV(T,F) = \int \mathrm{IF}(z;\ T,F)^2\,dF(z) \qquad (2.25)$$

is the asymptotic variance. A rigorous derivation is given in [36, 37]. If F has a density f with respect to some measure then by the asymptotic Cramer-Rao inequality,

$$AV(T,F) \geq \left[I(F_\theta)\right]^{-1} \qquad (2.26)$$

where $I(F)$ is the Fisher information.

[38] showed that the asymptotic Cramer-Rao lower bound is achieved by any Fisher consistent functional T with finite influence function if and only if

$$\mathrm{IF}(z;\ T,F) = \left[I(F_\theta)\right]^{-1} \frac{\partial}{\partial\theta}(\ln f_\theta). \qquad (2.27)$$

2.4 Robustness measures

2.4.1 Breakdown point

The breakdown point is a global measure of robustness. It is the smallest proportion of contamination that can cause the estimator to `break down' and to get arbitrarily far from the real relationship. To give mathematical definition of the finite breakdown point, let us define the maximum effect (bias).

Definition 2.8: Let *T* as a regression estimator, $\mathbf{Z} = \{(x_{11},\dots, x_{1p}, y_1), \dots, (x_{11},\dots, x_{np}, y_n)\}$ as a sample of *n* observations, such that $T(\boldsymbol{Z}) = \hat{\boldsymbol{\beta}}$. Let ***Z′*** *be* **Z** with m of the n observations corrupted. The maximum effect (bias) caused by such contamination is

$$bias\left(m;\ T,\ \boldsymbol{Z}\right) = sup_{Z}\ \|T\left(\boldsymbol{Z}'\right) - T\left(\boldsymbol{Z}\right)\| \tag{2.28}$$

where the supremum is over all possible Z'. If bias (m; T, Z) is infinite, *m* outliers can have an arbitrarily large effect on *T*.

Definition 2.9: The finite breakdown point of *T* at the sample ***Z*** is therefore defined as:

$$\epsilon_n^*\left(T,\ \boldsymbol{Z}\right) = min\{\tfrac{\mathrm{m}}{\mathrm{n}} : bais\left(m; T,\ \boldsymbol{Z}\right) = \infty\}\ [26]. \tag{2.29}$$

One outlying observation can offset the OLS estimate, so it has ϵ_n^* of $\frac{1}{\mathrm{n}}$. When $n \to \infty$, then *T* has a breakdown point of 0%, means that only a single outlying observation can cause an estimator to be meaningless. High breakdown points as $\frac{\mathrm{n}}{2}$ (or 50%), means that if we have contaminated data up to half, the estimator can still be useful.

2.4.2 Gross-Error Sensitivity

The gross-error sensitivity can be used to quantify the robustness of an estimator T at a distribution F. Recall IF (z; T, F) means that an infinitesimally small amount of outliers can have an arbitrary large effect on the result. This motivates the definition of the gross error sensitivity (GES) for T at F as the supremum of the influence function of an estimator. It can be defined as

$$\gamma^* = \underset{\mathrm{z}}{\mathrm{sub}}\left|\mathrm{IF}\left(\mathrm{z};\ \mathrm{T},\ \mathrm{F}\right)\right|. \tag{2.30}$$

The GES thus means that the worst influence an infinitesimally small fraction of contamination can have on the estimate, i.e. it is an upper bound for the standardized bias induced by the contaminated distribution. If the gross-error sensitivity is unbounded, $\gamma^* = \infty$, then the estimator is completely uncharitable of outliers; a single outlier can breakdown the estimator. For robust estimators, the supremum of its influence function should be bounded, then an infinitesimally small amount of contamination introduced in the data cannot cause considerably changes in estimation and a certain degree of robustness is indeed present.

2.5 Robust Regression methods

Many of popular modern robust regression techniques perform well in terms of the properties given above. The following subsections illustrate the most important being least Absolute Values estimation [39], M-estimates proposed [40], Least Trimmed Squares (LTS) estimates [41] and MM-estimates [25] for robust estimation.

2.5.1 L-norm or Least Absolute Values Estimation

Edgeworth in [39] has proven that squaring of the residual causes to OLS to become extremely vulnerable to the presence of outliers. Thus the proposed the L_1-norm or least absolute values (LAV) regression estimator minimizes the sum of the absolute values of the residuals rather than the sum of their squares. In fact, the least absolute value (LAV) is a robust maximum likelihood estimator suitable for the heavy-tailed Laplace distribution, which defines the L_1, or median regression, estimator as

$$\underset{\beta \in \Re^p}{\operatorname{argmin}} \sum_{i=1}^{n} |\mathrm{r_i}| = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum_{i=1}^{n} \left| y_i - \boldsymbol{x}_i' \hat{\boldsymbol{\beta}} \right| \tag{2.31}$$

Although LAV is less affected than OLS by vertical outliers, it fails to account for leverage and thus has a breakdown point of $\epsilon_n^* = 0$ [42]. Therefore, one outlying observation can totally offset the LAV estimator. However, LAV estimates have relatively low efficiency, about 64% efficiency. Thus, the combination of the low breakdown point and low efficiency makes LAV less attractive than other robust regression methods.

2.5.2 M-Estimation

M-estimators are maximum likelihood estimators, which were first, proposed by [43] for estimating location then were extended in relatively straightforward for regression [40]. They consider a tradeoff between the efficiency of the least squares estimators and the robustness of the LAV estimators.

The M-estimators try to down-weight the effect of the outliers by replacing the squared residuals $\mathrm{r_i^2}$ by a less rapidly increasing the residuals function $\rho(\mathrm{r})$, this function is related to the likelihood function for an suitable choice of the error distribution, yielding

$$\hat{\boldsymbol{\beta}}_{\mathrm{M}} = \underset{\beta\in\Re^{\mathrm{p}}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(\mathrm{r}_{\mathrm{i}}) = \underset{\beta\in\Re^{\mathrm{p}}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(y_{i} \boldsymbol{-x}_{i}'\hat{\boldsymbol{\beta}}) \tag{2.32}$$

where function ρ gives the contribution of each residual to the objective function. A reasonable ρ should have the following properties:

1) always-non negative, $\rho(\mathrm{r}) \geq 0$,
2) equal to 0 when its argument is 0, $\rho(0)= 0$,
3) symmetric, $\rho(r) = \rho(-r)$, although in some problems one might argue that symmetry is undesirable.

From now on ρ will denote a ρ-function. Setting $\rho(\mathrm{r}) = \frac{1}{2}\mathrm{r}^{\frac{1}{2}}$ we get the OLS estimator. This illustrates that the least squares estimator is a special case of the ρ-function satisfies these requirements, but it is not particularly robust. A possible choice is to set ρ $(r) = |r|$. This gives us the median regression (L_1-estimator, LAV, or LAD).

Unfortunately, M-estimators are not scale equivariant even if they are regression equivariant. Hence, the minimization problem is modified by dividing the ρ-function by a robust estimate of scale $\hat{\sigma}$, so the formula (2.32) becomes

$$\hat{\boldsymbol{\beta}}_{\mathrm{M}} = \underset{\beta\in\Re^{\mathrm{p}}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{\mathrm{r}_{i}}{\hat{\sigma}}\right) = \underset{\beta\in\Re^{\mathrm{p}}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{y_{i} \boldsymbol{-x}_{i}'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right) \tag{2.33}$$

Among all the possible estimations of the standard deviation σ of the data, there is no rule how to choose it. A popular choice (e.g. [44], [45]) is the Median Absolute Deviation (MAD)

$$\hat{\sigma} = \mathsf{C} \ \operatorname{median}\left|\mathsf{r}_{i} - \operatorname{median}\left(\mathsf{r}_{i}\right)\right| \tag{2.34}$$

where C is a correction factor which depends on the distribution. For example C = 1.4826 is used to make $\hat{\sigma}$ an asymptotically unbiased estimator of σ and the sample actually arises from a normal distribution.

From formula (2.34), the Median Absolute Deviation (MAD) is computationally efficient. It is very robust as witnessed by its bounded influence function and its 50% breakdown point but it has a low (37%) normal efficiency [46].

For a convex ρ, equivalence to (2.33) can be found by finding the first partial derivatives of (2.32) with respect to $\hat{\boldsymbol{\beta}}$ and setting the result equal to zero, i.e. $\hat{\boldsymbol{\beta}}_{\mathrm{M}}$ is the solution of

$$\min\sum_{i=1}^{n}\psi\left(\frac{\mathrm{r_i}}{\hat{\sigma}}\right)=\min\sum_{i=1}^{n}\psi\left(\frac{y_i\ \boldsymbol{-x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}\right)\boldsymbol{x}_i=0 \tag{2.35}$$

where $\psi(r)=\rho'(r)$ is called the influence score or score function, resulting in the necessary condition normal equations. If $\psi(r)$ = r, then (2.35) reduces to the normal equations yielding the least squares estimator. If now we define the weight function w_i = $w(\mathrm{r_i})$ = $\psi(\mathrm{r_i})/\mathrm{r_i}$, then solution of equation (2.35) becomes

$$\hat{\boldsymbol{\beta}}_\mathrm{M}=\underset{\beta\in\Re^\mathrm{p}}{\mathrm{argmin}}\sum_{i=1}^{n}w\ \left[\left(y_i\ \boldsymbol{-x}_i'\hat{\boldsymbol{\beta}}\right)/\hat{\sigma}\right]\ \left(y_i\ \boldsymbol{-x}_i'\hat{\boldsymbol{\beta}}\right)/\hat{\sigma}=0. \tag{2.36}$$

Then we obtain a nonlinear system of equations. Iteratively reweighted least squares (IRLS) proposed by [47] is the most common nonlinear optimization technique in robust regression. Then the solutions of (2.36) can be expressed as:

$$\mathbf{X'WX}\hat{\boldsymbol{\beta}}\ \mathbf{=X'Wy} \tag{2.37}$$

where **W** is an n x n diagonal matrix of weights with diagonal elements $w_1, w_2, ..., w_n$ given by

$$w_\mathrm{i}=\frac{\psi\left[\left(y_i\ \mathbf{-x}_i'\hat{\boldsymbol{\beta}}\right)/\hat{\sigma}\right]}{\left(y_i\ \mathbf{-x}_i'\hat{\boldsymbol{\beta}}\right)/\hat{\sigma}} \tag{2.38}$$

The equation in (2.37) results in the usual weighted least squares normal equations. Thus, the one-step M-estimator can be found at convergence, where

$$\hat{\boldsymbol{\beta}}=\mathbf{(X'WX)}^{-1}\mathbf{X'Wy} \tag{2.39}$$

The following algorithm explains iteratively reweighted least squares or IRLS to obtain M-estimator $\hat{\boldsymbol{\beta}}_M$.

1) Select initial estimates $\hat{\boldsymbol{\beta}}_0$, such as the least-squares estimates.

2) calculate the estimate $\hat{\sigma}$ of σ

   (e.g. MAD: $\hat{\sigma}$ = 1.4826 $\text{median}\left|\mathrm{r}_i-\text{median}\left(\mathrm{r}_i\right)\right|$)

3) At each iteration j, with $\hat{\sigma}$ remains fixed throughout, calculate residuals $r_i^{(j-1)}$ and associated weight $w\ \left(r_i^{(j-1)}\right)$ according to the weight function.

4) Solve the following for re-weighted least squares (IRLS) equation,

$$\hat{\boldsymbol{\beta}}_1^{(j-1)} = (\mathbf{X}'\mathbf{W}^{(j-1)}\mathbf{X})^{-1}(\mathrm{X}'\mathbf{W}^{(j-1)}\mathbf{y}), \tag{2.40}$$

Steps 3) and 4) are repeated until $\dfrac{\left|\boldsymbol{\beta}_1^{(j)} - \boldsymbol{\beta}_1^{(j-1)}\right|}{\boldsymbol{\beta}_1^{(j-1)}}$ becomes less than tolerance.

The ψ-functions controls the weight given to each residual and is very important in determining the robust and efficiency properties of the estimator. Several suggestions for ψ-functions, have been made in the literature, they primarily belong to one of two categories: monotonic and redescending. The ψ-function is unbounded meaning large residuals receive heavy weights. The Huber function [43] is an example of a monotone ψ-function which results in down-weighting the large residuals compared to least squares. Huber function always provides the global solution regardless of the initial estimate. Other ψ-functions redescend with increasing residual magnitude. A popular example is the bisquare or biweight ψ-function suggested by [47].

Table 2.1 Objective functions and score functions for ordinary least-squares, Huber, and biweight estimators.

Method	**Objective Functions**	**Score Functions**
Least-Squares	$\rho_{OLS}(r) = r^2/2$	$\psi_{OLS}(r) = r$
Huber	$\rho_{HU}(e)=\begin{cases} r^2/2 & \text{if } \lvert r\rvert \le k_{HU} \\ k_{HU}\lvert r\rvert - k_{HU}^2/2 & \text{if } \lvert r\rvert > k_{HU} \end{cases}$	$\psi_{HU}(r)=\begin{cases} r & \text{if } \lvert r\rvert \le k_{HU} \\ \text{sgn}(r).\,r & \text{if } \lvert r\rvert > k_H \end{cases}$ $= \min\{\max\{-k_{HU}, r\}, k_{HU}\}$
Biweight	$\rho_{BI}(r)=\begin{cases} \dfrac{k_{BI}}{6}\left[1-(1-(r/k_{BI})^2)^3\right] & \text{if } \lvert r\rvert \le k_{BI} \\ \dfrac{k_{BI}}{6} & \text{if } \lvert r\rvert > k_{BI} \end{cases}$	$\psi_{BI}(r)=\begin{cases} (1-(r/k_{BI})^2)^2 & \text{if } \lvert r\rvert \le k_{BI} \\ 0 & \text{if } \lvert r\rvert > k_{BI} \end{cases}$

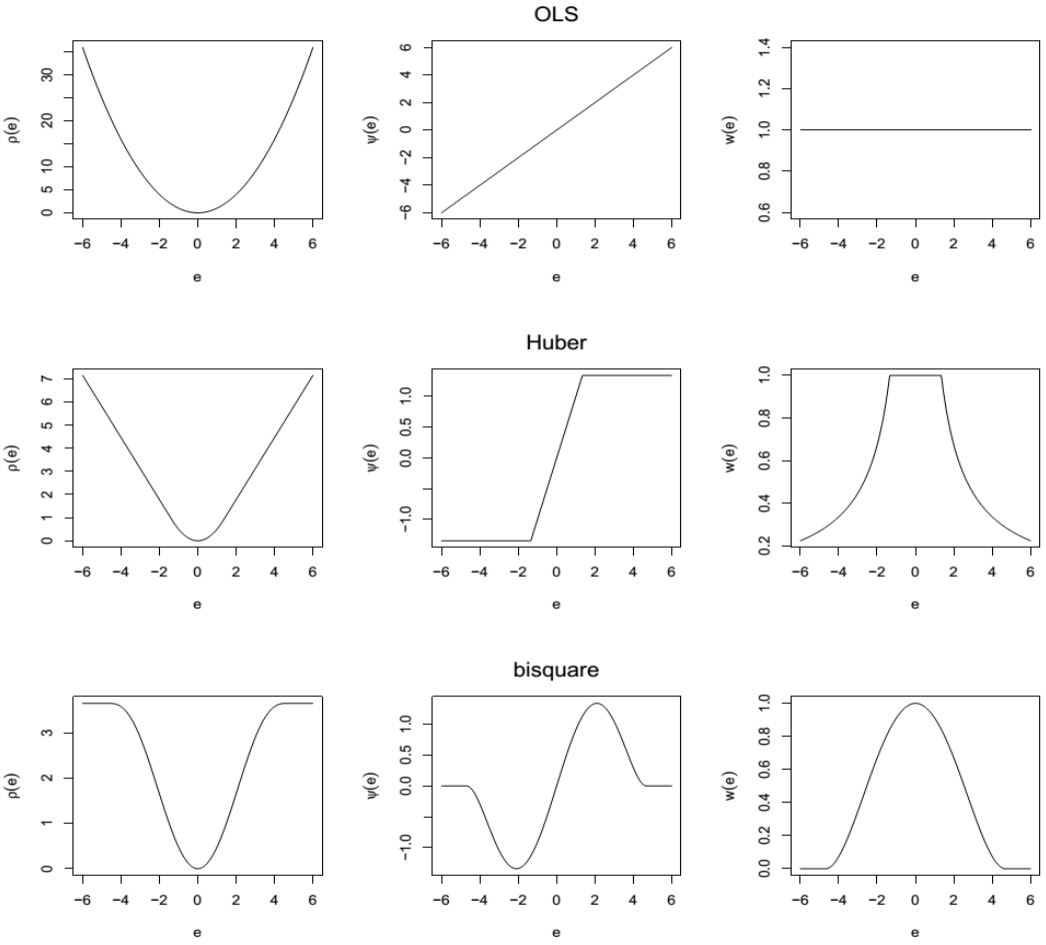

Figure 2.2 Objective (left), score (center), and weight (right) functions for the least-squares (top), Huber (middle), and bisquare (bottom) estimators. The tuning constants for these graphs are for the Huber estimator and for the bisquare [48].

Although redescending M-estimators can achieved better results in terms of efficiency and robustness that completely ignore large outliers, there are local minima in the optimization problem. In such cases, it is usual to choose a good starting point and iterate carefully. Table 2.1 presents three of score functions together with the corresponding objective functions. The objective functions, and the corresponding and weight functions for the three estimators are also given in Figure 2.2.

In fact each set of ψ-functions includes tuning constants, which have standard values resulting in estimators with desired efficiencies. For example the values k_{HU} =1.345 and k_{BI} =4.685 for the Huber and Tukey ψ-functions each achieve 95% asymptotic relative efficiency. However, the higher the efficiency of the M-estimator, the higher the maximum bias due to data contamination. [49] recommend the bisquare ψ-function with efficiency at the normal set to 85%.

2.5.2.1 Distributional properties of M-estimation

It is interested to investigate in the distributional properties of $\hat{\boldsymbol{\beta}}_M$. In the situation where the number of regression parameters p is fixed and the number of observations n is tending to infinity, [35] and [40] showed that, under certain conditions, the asymptotic distribution of $\hat{\boldsymbol{\beta}}_M$ is N ($\boldsymbol{\beta}$, AV), where the asymptotic variance matrix AV is given by

$$\mathrm{AV} = \mathrm{V}(\psi, \mathrm{F})(\mathbf{X}'\mathbf{X})^{-1} \tag{2.41}$$

and ψ is continuous, bounded and has a bounded derivative and V(ψ, F) = $(\int \psi^2 \mathrm{dF}) \big/ \left(\int \psi' \ \mathrm{dF}\right)^2$ the asymptotic variance functional of [50], and ($\mathbf{X}'\mathbf{X}$) the usual Gram matrix associated with the least squares problem. [51] pointed out that the smallest possible asymptotic variance when ψ(x) = $\left(\log f\ (x)\right)'$ (with f the density of errors term), the asymptotic variance yields V(ψ, F) = 1/I(F), with I(F) denoting the Fisher information. I.e. the optimal M-estimator depended on the probability distribution F.

Unfortunately, the finite sample distribution of $\boldsymbol{\beta}$ and its covariance matrix is not known. [51] point out that one approach to robust inferential procedures based on $\boldsymbol{\beta}$ utilizes finite sample approximations to AV. They discuss several alternative finite sample estimates of the covariance matrix of $\boldsymbol{\beta}$.

2.5.2.2 Influence function of M-estimator

T(F) for M-estimators in regression is defined implicitly in:

$$\int \mathrm{X}'\psi\left(\mathrm{y} - \mathrm{XT(F)}\right)\mathrm{dF} = 0 \tag{2.42}$$

The corresponding influence function for M-stimators is then

$$\mathrm{IF}(\mathbf{z}, T, \mathrm{F}) = \left(\psi(\mathrm{r}) \Big/ \int \psi'(\mathrm{r})\mathrm{d}\Phi\right).\left(\int \mathrm{X}'\mathrm{X}\mathrm{dK}\right)^{-1} \mathrm{X} \tag{2.43}$$

where $\mathbf{r} = \mathbf{y} - X\hat{\boldsymbol{\beta}}$, Φ the normal distribution of the errors, and K is the distribution function of $\mathbf{x}_i$. There are two components of the IF, the influence of the residual (IR) and the influence of position (IP) such that

$$\text{IF}(\mathbf{X}, X\hat{\boldsymbol{\beta}} + \mathbf{r};\text{T},\text{F}) = \text{IR}(\mathbf{r};\text{T},\Phi).\text{IP}(\mathbf{X};\text{T},\text{K}) \tag{2.44}$$

For an estimator to have bounded influence, both the IR and IP must be bounded. Thus, M-estimators have bounded IR if y is bounded, but unbounded IP. Thus the influence function with respect of $\boldsymbol{y}$ can be bounded by choice of ψ, but the influence function of M-estimators is unbounded in respect of $\mathbf{X}$. Thus, the IFs for the cases of monotone and of redescending ψ are rather different. If ψ is monotone, then the IF tends to infinity for any fixed $\mathbf{x}_0$ if y_0 tends to infinity. If ψ is redescending and is such that $\psi(x) = 0$ for $|x| \geq k_{BI}$, then the IF will tend to infinity only when $\mathbf{x}_0$ tends to infinity and $|y_0 - \mathbf{x}_0\boldsymbol{\beta}|/\sigma \leq k_{BI}$, which means that large outliers have no influence on the estimate. It follows that the IF of M-estimators is unbounded and have an overall breakdown point of 0%. I.e. a single outlier can have an unbounded effect on the M-estimator.

2.5.3 Least Trimmed Squares (LTS) Estimation

Least trimmed Squares (LTS) estimation is a high breakdown value method introduced by [41] as a high efficiency alternative to least median squares (LMS). The least trimmed squares estimator is found by

$$\hat{\boldsymbol{\beta}}_{\text{LTS}} = \underset{\beta \in \Re^{\text{p}}}{\text{argmin}} \sum_{i=1}^{\text{h}} \left(\text{r}^2(\boldsymbol{\beta})\right)_{i:n} \tag{2.45}$$

where $\left(\text{r}^2(\boldsymbol{\beta})\right)_{1:n} \leq \ldots \leq \left(\text{r}^2(\boldsymbol{\beta})\right)_{n:n}$ are the order statistics of the squared residuals, the so-called trimming constant $h = \lfloor n(1-\alpha)+1 \rfloor$ is the number of observations included in the calculation of the estimator, and α is the proportion of trimming that is performed. LTS estimator fits the best subset of h observations, removing n – h observations from the sample even they are not outliers. Thus, the estimator has high breakdown point but loses efficiency.

In fact, LTS has important statistical properties. First, the least trimmed squares estimator is regression, scale, and affine equivariant [25]. The maximum breakdown point reaches $\left(\left[(n-p)/2\right]+1\right)/n$ when $h = \lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$. In the situation where the number of regression parameters p is fixed and the number of samples n is tending to infinity, h is approximately $n/2$, in which case the breakdown point attains 50%. Whereas for $h = n$, which corresponds to the least squares estimator, the breakdown point equals to 0. Furthermore, for a general linear regression model, the LTS-estimator is asymptotically normal with rate of convergence $n^{1/2}$ under some conditions, but it has poor asymptotic efficiency under normal errors [52]. Instead, the LMS estimator converges to a nonnormal distribution with rate of convergence $n^{1/3}$ [53].

Although LTS has good statistical properties, in real applications, the exact computation of LTS is difficult. It is equivalent to searching for subset of h observations whose least squares fit produces the smallest sum of squared residuals. Given subset size h as an initial guess of the number of good observations in the data, the minimization of the objective function in (2.45) can be viewed as a process in which every time choosing subsample of h observations and find some $\boldsymbol{\beta}$ minimizing the sum of squared residuals for the selected subsample. The final estimate is the one that commands the smallest value of the objective function. The total number of size-h subsets in a sample of size n is $\binom{n}{h}$ subsets, full search through all size-h subsets is impossible unless the sample size is small.

Instead of exhaustive search, fast probabilistic algorithms have been developed to compute approximate solutions for larger samples. These algorithms based on random search for sample size-h subsets. Alternatively, so-called C-step technique (C stands for "concentration") proposed by [54], can be used to improve the objective function (2.45). This step can be iterated starting from any subset and is applied to all candidates obtained by subsampling. It makes each candidate closer to the solution of the optimization problem. If the C-step is applied a sufficient (finite) number of times, a local minimum of the objective function is obtained.

The fast-LTS algorithm, developed by [55] proposed a modification of the subsampling algorithm for the LTS-estimator using fixed number k of C-steps. [55] recommended to take k = 2. They also showed that the fast-LTS is that considerably improves its performance and becomes much faster than the approximating algorithms for the LTS-estimator that do not use the C-step.

2.5.4 MM-estimation

MM estimation, which was introduced by [25], combines high breakdown value estimation and M estimation. It has both the high breakdown property and a higher statistical efficiency than LTS estimation.

First we want to define a scale M-estimate $\hat{\sigma}_M$, which is a robust scale of the residuals $r_i(\hat{\boldsymbol{\beta}}_0)$.

Let ρ_0 a bounded ρ-function, the scale M-estimate (an M-scale for short) $\hat{\sigma}$[43], satisfies

$$\frac{1}{n-p}\sum_{i=1}^{n}\rho_0\left(\frac{r(\beta)}{\hat{\sigma}}\right)=b\,. \tag{2.46}$$

where b $\in$ (0, 1) is a constant that controls the estimates robustness and satisfies b = E_Φ [ρ (t^*)], with Φ the standard normal distribution.

There are three stages that define an MM-estimator [25]:

1. Compute an initial consistent estimate $\hat{\beta}_0$ with high breakdown but possibly low normal efficiency (typically LTS or S-estimator proposed by [56] are used).
2. Compute the residuals

   $r_i(\hat{\beta}_0) = y_i - \boldsymbol{x}_i'\hat{\beta}$, for $1 \le i \le n$.

   and compute the M-scale $\hat{\sigma}_M = \hat{\sigma}(r_i(\hat{\boldsymbol{\beta}}))$ defined by (2.40) using a bounded ρ-function $\rho_0(r) = \rho(r/k_0)$ and b = 0.5, this due to the asymptotic breakdown of $\hat{\sigma}_M = 0.5$.

3. Let $\rho_1(r) = \rho(r/k_1)$ be another bounded ρ-function, such that $\rho_1(r) \le \rho_0(r)$, then the MM-estimate $\hat{\boldsymbol{\beta}}_1$ is defined as a solution of (2.30) using an iterative procedure starting at $\hat{\boldsymbol{\beta}}_0$.

If $\hat{\boldsymbol{\beta}}_1$ is such that

$$L(\hat{\boldsymbol{\beta}}_1) \le L(\hat{\boldsymbol{\beta}}_0)$$

where

$$L(\hat{\boldsymbol{\beta}}_1)=\sum_{i=1}^{n}\rho(\frac{y_i\ \boldsymbol{-x}_i'\hat{\boldsymbol{\beta}}}{\hat{\sigma}}) \tag{2.47}$$

then $\hat{\boldsymbol{\beta}}_1$ is consistent and its ϵ_n^* is not less than that of $\hat{\boldsymbol{\beta}}_0$. If furthermore $\hat{\boldsymbol{\beta}}_1$ is any solution of (2.35), then it has the same efficiency as the global minimum [25].

The value of k_0 should be chosen in order to attain high breakdown point of the MM-estimation. The choice of k_1 will to determine asymptotic efficiency of the estimate without affecting its breakdown point. In order to let $\rho_1 \le \rho_0$, we must have $k_1 \ge k_0$; the larger the k_1 is, the higher efficiency the MM-estimation can attain at the normal distribution.

CHAPTER 3

REGULARIZED LEAST SQUARES REGRESSION METHODS

The OLS estimator is the best-unbiased estimator (BLUE) if the predictors are orthogonal. However, for high dimensional data, where the number of predictors are potentially much larger than the number of samples, things can go wrong. If p is large, some of the columns of $\mathbf{X}$ are likely to be nearly collinear, making $\mathbf{X}'\mathbf{X}$ 'almost singular'. Then the inverse $(\mathbf{X}'\mathbf{X})^{-1}$ will have some very large eigenvalues, and consequently the OLS estimator exhibits a very large variance, var ($\hat{\boldsymbol{\beta}}_{\text{OLS}}$) = $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. In addition, the length of the OLS vector tends to be longer than the length of the true parameter vector, i.e. $\|\hat{\boldsymbol{\beta}}_{\text{OLS}}\| \geq \|\hat{\boldsymbol{\beta}}_{\text{True}}\|$ [1].

To overcome these problems regularization techniques was proposed by [1] for regression models. The regularization methods are based on penalty terms and should yield unique estimates of the parameter vector $\boldsymbol{\beta}$. Furthermore, an improvement of fitting in terms of prediction error rates by sacrificing some bias to reduce the variability in the estimates of regression coefficients. For example lasso, elastic net shrink some coefficient estimates to exactly zero, thus, much like best subset selection, they perform variable selection. Thereby we obtain regression models which should contain only the strongest effects and which are easier to interpret.

Regularized least squares (RLS) estimation provides a way to regularize fitting the data. The estimation of regression coefficients are obtained by minimizing an objective function that involves a penalty function on top of the sum of squared residuals. Hence the classical regression model is given by (2.7).

Without loss of generality, we can assume that the predictors are all standardized (i.e. all the predictors have mean 0 and variance 1), and the response variable has mean 0. Consequently:

$$\frac{1}{n}\sum_{i=1}^{n} y_i = 0,\quad \frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0;\quad \frac{1}{n}\sum_{i=1}^{n} x_{ij}^{2} = 1 \ \ \forall j \in \{ 1,\ldots,\mathrm{p}\}.$$

Then the regularized least squares estimates are obtained by minimizing

$$RLS\,(\lambda,\boldsymbol{\beta}) = (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \sum_{\mathrm{j}=1}^{\mathrm{p}} \mathrm{P}_{\lambda}\left(\left|\beta_{\mathrm{j}}\right|\right) \tag{3.1}$$

The penalty function $\mathrm{P}_{\lambda}(.)$ depends on an unknown strictly positive thresholding parameter λ. This penalty parameter λ controls the shrinkage intensity. For the tuning parameter $\lambda = 0$ we obtain the ordinary least squares solution. On the contrary, for large values of λ the influence of the penalty term on the coefficient estimates increases.

Many classical model selection criteria have been applied in order to choose λ in the sense of the prediction, such as Akaike Information Criterion (AIC [3]), Bayesian Information Criterion (BIC [4]), Mallows C_p [5], and Generalized Cross-Validation (GCV [57]) as well as k-fold cross-validation methods. More detail in the next section:

3.1 Model Assessment and Selection

For all regularized least squares (RLS) methods, we can think of an estimate

$$\hat{\boldsymbol{y}} = \hat{f}\,(\mathbf{X}) \tag{3.2}$$

where $\hat{f}$ represents the estimate f, and $\hat{\boldsymbol{y}}$ represents the resulting prediction for $\boldsymbol{y}$. Furthermore, for all of these methods, let $\hat{f}_{\lambda}$ be the estimate indexing with parameter λ.

There are two goals when solving a prediction problem: model selection and model assessment. The model selection is estimating the performance of different models in order to choose the best one. Model Assessment is having chosen the final model, estimating its prediction error on new data. Thus, the best model is defined as the one with the lowest Expected prediction error (EPE):

$$\begin{aligned}\mathrm{EPE}(\lambda) = \mathrm{E}(\boldsymbol{y}-\hat{\boldsymbol{y}})^2 &= \mathrm{E}[f\,(\mathbf{X}) + \boldsymbol{\varepsilon} - \hat{f}_{\lambda}(\mathbf{X})]^2 \\ &= \underbrace{[f\,(\mathbf{X}) - \hat{f}_{\lambda}(\mathbf{X})]^2}_{Reducible} + \underbrace{\mathrm{Var}(\boldsymbol{\varepsilon})}_{Irreducible}\ ,\end{aligned} \tag{3. 3}$$

where $\mathbf{y}$ and $\boldsymbol{x}$ are drawn at random from the population.

The accuracy of $\hat{\boldsymbol{y}}$ as a prediction for y depends on two types of error: the reducible error and the irreducible error. In general, $\hat{f}$ will not be a perfect estimate for f, and this inaccuracy will introduce some error. This error is reducible because of the improving the accuracy of $\hat{f}$ depends on model selection. The reducible error term is also referred to as model error. The noise error term also affects the accuracy of the predictions. This is known as the irreducible error, because it that cannot fundamentally be reduced by model choice.

When building prediction model, the main goal should be making a model that most accurately predicts the desired target value for a given data set. I.e. the predicted response value for a given observation is close to the true response value for that observation. Thus, we need to measure of model prediction error. In the regression setting, the most commonly-used measure is the mean squared error (MSE), given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\boldsymbol{x}_i))^2, \tag{3.4}$$

where $\hat{f}(\boldsymbol{x}_i)$ is the prediction that $\hat{f}$ gives for the ith observation. The MSE in (3.4) is computed using the training data that was used to fit the model, so it is called the training MSE.

3.1.1 Bias-Variance trade-off

Let f_λ be the estimate obtained with the training data. If we have a completely independent data point, let's simplify by assuming $\mathbf{X} = x*$ is fixed. So our goal is to minimize

$$\begin{aligned}\mathrm{EPE}(\boldsymbol{x}_0) &= \mathrm{E}[y^* - \hat{f}_\lambda(\boldsymbol{x}^*)]^2 \\ &= \sigma^2 + \{\mathrm{E}[\hat{f}_\lambda(\boldsymbol{x}^*)] - f(\boldsymbol{x}^*)\}^2 + \mathrm{var}[\hat{f}_\lambda(\boldsymbol{x}_0)]\end{aligned} \tag{3.5}$$

$$= \text{Irreducible error} + \text{Bias}^2 + \text{Variance}$$

In general, we want to choose a λ that performs well for all $\boldsymbol{x}$. If instead of just one new point we obtain n then we would have

$$\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}[y_i^* - \hat{f}_\lambda(\boldsymbol{x}_i^*)]^2 = \sigma^2 + \{\mathrm{E}[\hat{f}_\lambda(\boldsymbol{x}_0)] - f(\boldsymbol{x}_0)\}^2 + \mathrm{var}[\hat{f}_\lambda] \tag{3.6}$$

The notation (3.6) defines the overall expected test MSE that can be computed by averaging $\mathrm{E}[y_i^* - \hat{f}_\lambda(x_i^*)]^2$ over all possible values of x_0 in the test set. Typically, there is a bias-variance tradeoff in choosing the appropriate complexity of the model.

3.1.2 Estimation of test error

Since the selection of the best model based on test error, so we want to estimate this test error. There are two common approaches to give an estimate of the test error [58]:

1. We can estimate test error indirectly by adjusting the training error to account for the bias due to overfitting (i.e. computing the training error and then adjusting it).
2. We can directly estimate the test error by fit models on part of the data, and then evaluate them on a holdout set.

3.1.2.1 Model selection criteria C_p, AIC and BIC

These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.

Mallows C_p is defined as

$$C_p = n^{-1}\left(\sum_{i=1}(y_i - \hat{y}_i)^2 + 2\cdot\text{df}(\hat{\mathbf{y}})\hat{\sigma}\right), \tag{3.7}$$

Akaike Information Criterion AIC is defined as

$$\text{AIC} = n\log(n^{-1}\sum_{i=1}(y_i - \hat{y}_i)^2) + 2\cdot\text{df}(\hat{\mathbf{y}}), \tag{3.8}$$

and Schwarz BIC-score can be defined as

$$\text{BIC} = n\log(n^{-1}\sum_{i=1}(y_i - \hat{y}_i)^2) + \log(n)\cdot\text{df}(\hat{\mathbf{y}}), \tag{3.9}$$

where $\hat{\sigma}$ is an estimate of the variance of the error ε in the classical linear (2.7), and $(\hat{\text{y}}_1,\ldots,\hat{\text{y}}_\text{n})'$ is the predicted response and $\text{df}(\hat{\mathbf{y}})$ denotes the degree of freedom of the fitted model.

3.1.2.2 The Validation Set Approach

The validation set approach, is a very simple strategy for estimate the test error. Here randomly dividing the available set of observations into two parts: a training set and a validation or hold-out

The model is fitted on the training set, and the fitted model is used to predict the responses for the observations in the validation set. The resulting validation-set error provides an estimate of the test error. Typically, the performances of the models are measured using MSE.

Although validation set is simple and easy to implement, it has two potential drawbacks:

1. The validation estimate can be highly variable and depend highly upon which observations are included in the training set and which observations are included in the validation set.
2. In the validation approach, only a subset of observations is used to fit the model (training data set). This suggests that the validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

3.1.2.3 Leave-One-Out Cross-Validation

In practice, it is not common to have a new set of data or the number of observations is often limited. Therefore, it often advisable to more carefully utilizes available observations. Cross validation is an approach to do just that.

Leave-one-out (LOOCV) cross-validation attempts to address drawbacks of the validation set. LOOCV involves splitting the set of observations into two parts. However, instead of creating two subsets of comparable size, a single observation ($\boldsymbol{x}_1$, y_1) is used for the validation set, and the n-1 remaining observations make up the training set. Since ($\boldsymbol{x}_1$, y_1) was not used in the fitting process, $\text{MSE}_1 = (y_1 - \hat{y}_1)^2$ provides an approximately unbiased estimate for the test error. Even though the validation set is an unbiased test error, it could be highly variable, since it is based on a single observation ($\boldsymbol{x}_1$, y_1).

To remedy this, we repeat the procedure *n* times by alliteratively leaving one observation out, producing n squared errors, $\text{MSE}_1, \ldots, \text{MSE}_n$ and then computing the average MSE of all *n* test estimates which equal cross-validation (CV):

$$CV(n) = \frac{1}{n}\sum_{i=1}^{n} MSE_i = n^{-1}\sum_{i=1}^{n}\{y_i - \hat{f}_\lambda^{-i}(\boldsymbol{x}_i)\}^2 \qquad (3.10)$$

where $\hat{f}_\lambda^{-i}(\boldsymbol{x}_i)$ indicates the fit at $\boldsymbol{x}_i$ computed by leaving out the ith point.

The CV (λ) is computed for a grid of values of λ for choosing the λ that minimizes it.

3.1.2.4 Validation k-Fold Cross-Validation

K-fold approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k - 1$ folds. The mean squared error, MSE$_1$, is then computed on the observations in the held-out fold. This procedure is repeated *k* times; each time, a different group of observations is treated as a validation set. This process results in *k* estimates of the test error, MSE$_1$, MSE$_2$, ..., MSE$_k$. Then the *k*-fold cross-validation estimate of the expected test error is computed by averaging these values,

$$CV\,(\lambda) = \frac{1}{k}\sum_{i=1}^{k} MSE_i \tag{3.11}$$

Or can given in this formula as [59]

$$CV\,(\lambda) = n^{-1}\sum\nolimits_{k=1}^{K}\sum\nolimits_{i\in C_k}\left\| y_i - \hat{f}_{\lambda}^{-k}(\boldsymbol{x})\right\|^2. \tag{3.12}$$

In fact, LOOCV is a special case of *k*-fold CV in which *k* is set to equal *n*. In practice, one typically performs *k*-fold CV using *k* = 5 or *k* = 10. The most advantage of using *k* = 5 or *k* = 10 rather than *k* = *n* is computational speed. LOOCV requires fitting *n* times. This has the potential to be computationally expensive. So performing LOOCV may pose computational problems, especially if *n* is extremely large. In contrast, performing 10-fold CV requires fitting only ten times. The difference between a 5, 10, *n* or other sized *k*-folds CV is the bias-variance tradeoff. However, generally, a 10-fold cross-validation will not be too different from a LOOCV.

3.2 Ridge regression

Ridge regression originally proposed by [1] is the most popular penalized least squares method that commonly used for dealing with multicollinearity as alternative solutions to OLS. The ridge regression parameter estimates are determined by minimizing the residual sum of squares subject L$_2$-penalty on the coefficients. Consequently the ridge estimator $\hat{\boldsymbol{\beta}}_{\text{Ridge}}$, is defined by

$$\hat{\boldsymbol{\beta}}_{Ridge} = \underset{\beta\in\Re^p}{\text{argmin}}\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2 \;, \quad s.t. \sum_{\text{j=1}}^{\text{p}}\beta_{\text{j}}^2 \le t,\;\; t \ge 0, \tag{3.13}$$

or equivalently, the ridge regression is defined by the following minimization penalized regression problem:

$$\hat{\boldsymbol{\beta}}_{Ridge} = \underset{\beta \in \Re^p}{\arg\min} \sum_{i=1}^{n} \left(y_i - x_i' \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \text{ , } \lambda \geq 0 \tag{3.14}$$

where $\lambda \geq 0$ is a regularization parameter that controls the amount of shrinkage and the penalty function is given by the L_2-norm: the larger the value of λ, the greater the amount of shrinkage (toward zero) i.e. the more you prefer the β_j 's close to zero. For examples, when parameter $\lambda = 0$ we obtain the ordinary least squares solution and when parameter $\lambda = \infty$, we have $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = 0$. There is a one to-one correspondence between the parameters λ in (3.13) and t in (3.14). This means that for a specific value *λ,* there exists a value *t* leads to the same solution.

Rewriting the equation (3.14) in matrix form yields,

$$\hat{\beta}_{Ridge} = \underset{\beta \in \Re^p}{\arg\min} = \left\| \mathbf{y} - \mathbf{X}\beta \right\|_2^2 + \lambda \left\| \beta \right\|_2^2 \tag{3.15}$$

To obtain a closed-form solution $\hat{\beta}_{Ridge}$ such as suggested by [1], the RSS for ridge regression is expressed as

$$RSS\left(\beta;\lambda\right) = \left(\mathbf{y} - \mathbf{X}\beta\right)'\left(\mathbf{y} - \mathbf{X}\beta\right) + \lambda\beta'\beta. \tag{3.16}$$

By using applications of matrix calculus, that is, setting to zero and taking the first derivative, we obtain

$$\frac{\partial}{\partial \boldsymbol{\beta}} RSS\left(\beta;\lambda\right) = 2\left(\mathbf{X'X}\right)\boldsymbol{\beta} - 2\mathbf{X'y} + 2\lambda\boldsymbol{\beta} = 0. \tag{3.17}$$

Equation (3.17) can be simplified as follows,

$$\begin{aligned} &2\left(\mathbf{X'X}\right)\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} = 2\mathbf{X'y}, \\ &\left(\mathbf{X'X} + \lambda\mathbf{I}\right)\boldsymbol{\beta} = \mathbf{X'y}, \end{aligned} \tag{3.18}$$

and therefore the ridge estimators are

$$\hat{\beta}_{\text{Ridge}} = (\mathbf{X'X} + \lambda\mathbf{I})^{-1}\mathbf{X'y} \tag{3.19}$$

where **I** is the p × p identity matrix. The solution adds a positive constant λ**I** to the diagonal of $\mathbf{X'X}$ producing an invertible matrix, $(\mathbf{X'X} + \lambda\mathbf{I})$. This makes the problem nonsingular, even if $\mathbf{X'X}$ singular. In addition, this also shows that $\hat{\boldsymbol{\beta}}_{\text{Ridge}}$ is still a linear function of the observed values, **y**.

Moreover, when **X** is composed of orthonormal variables, such that $\mathbf{X'X} = \mathbf{I}_p$, it then follows that

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{I}+\lambda\mathbf{I})-1\mathbf{X}'\mathbf{y} = ((1+\lambda)\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} = \frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}_{\text{OLS}}. \qquad (3.20)$$

In this simple case, the ridge estimator is simply a scaled version or a down-weighted version of the OLS estimator. The optimal choice of λ minimizing the expected prediction error is:

$$\lambda^* = \frac{\mathrm{p}\sigma^2}{\sum_{\mathrm{j}=1}^{\mathrm{p}} \beta_{\mathrm{j}}^2} \qquad (3.21)$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{\mathrm{p}})$ is the true coefficient vector.

3.2.1 The effective degrees of freedom

To find effective degrees of freedom for ridge regression, a linear matrix hat operator $\mathbf{H}_\lambda$ can be used that put the "hats" on y. Therefore, the fits are a linear combination of the y_i's,

i = 1, . . . , n, that satisfy:

$$\hat{\mathrm{y}} = \mathbf{H}_\lambda \mathrm{y} \qquad (3.22)$$

Then the effective degrees of freedom is trace hat matrix $\mathbf{H}_\lambda$

$$\mathrm{df}(\lambda) = \mathrm{tr}\left(\mathbf{H}_\lambda\right). \qquad (3.23)$$

However, there is a need to some additional insight into the spectral properties of **X**. The design matrix can be expressed using a singular value decomposition (SVD), such that

$$\mathbf{X} = \mathbf{UDV}', \qquad (3.24)$$

where **U** and **V** are orthogonal matrices of order n × p and p × p, respectively; and **D** is a p × p diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \cdot \cdot \cdot \geq d_p \geq 0$ called the singular values of **X**. If one or more values $d_j = 0$, **X** is singular. The columns of **U**, which are called the left singular vectors, span the column space of X; whereas the columns of **V**, which are called the right singular vectors, span the row space of **X**. In particular, **U** is a set of eigenvectors for $\mathbf{X}'\mathbf{X}$, and **V** is a set of eigenvectors for $\mathbf{X}'\mathbf{X}$.

Now, using the singular value decomposition SVD of **X** we can write the ridge regression fitted vector as

$$\mathbf{X}\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \mathbf{X}(\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} \qquad (3.25)$$

Hence, we can define hat matrix for ridge regression as

$$\mathbf{H}_{\lambda} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}' \tag{3.26}$$

Using the SVD of **X**, then the Gram matrix $\mathbf{X}'\mathbf{X}$ can be decomposed as follows,

$$\mathbf{X}'\mathbf{X} = \mathbf{VDU}'\mathbf{UDV}' = \mathbf{VD}^2\mathbf{V}' \ \mathbf{H}_{\lambda} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}' \tag{3.27}$$

which is the eigendecomposition of $\mathbf{X}'\mathbf{X}$. We can apply this to the hat matrix,

$$\begin{aligned} \mathbf{H}_{\lambda} &= \mathbf{UDV}'\left(\mathbf{VD}^2\mathbf{V}\mathrm{T}' + \lambda\mathbf{I}\right)^{-1}\mathbf{VDU}' \\ &= \mathbf{UDV}'\mathbf{V}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{V}'\mathbf{VDU}' \\ &= \mathbf{UD}\left(\mathbf{D}^2 + \mathbf{I}\right)^{-1}\mathbf{DU}'. \end{aligned} \tag{3.28}$$

since $\mathbf{VD}^2\mathbf{V}$ and $\lambda\mathbf{I}$ are simultaneously diagonalizable. Therefore, it also follows that $\mathbf{H}_{\lambda}$ is diagonalizable, with respect to **U**, and with eigenvalues given by $\mathbf{D}\left(\mathbf{D}^2 \ + \lambda\mathbf{I}\right)^{-1}\mathbf{D}$ which is a diagonal matrix of order $p \times p$. Thus, since the trace a matrix is equal to the sum of its eigenvalues, it follows that

$$\mathrm{tr}(\mathbf{H}_{\lambda}) = \sum_{\mathrm{j}=1}^{\mathrm{p}} \frac{d_{\mathrm{j}}^{2}}{d_{\mathrm{j}}^{2} + \lambda} \tag{3.29}$$

This means that the effective degrees of freedom of the ridge regression fit is monotone decreasing function of λ. When $\lambda = 0$, then

$\mathrm{df}(\lambda) = \ \mathrm{tr}\left(\mathbf{H}_{\lambda=0}\right) = \mathrm{p}$, there is no regularization is performed.

By contrast, if $\lambda \rightarrow \infty$, $\mathrm{df}(\lambda) \rightarrow 0$ Thus, regularization leads to a reduction in the effective number of parameters.

3.2.2 Bias and Variance of Ridge Estimator

Ridge estimation produces a biased estimator of the true parameter $\boldsymbol{\beta}$. Using the definition of $\hat{\boldsymbol{\beta}}_{\mathrm{Ridge}}$ and assumption of the linear model on mean function E[**y**|**X**] = $\mathbf{X}\boldsymbol{\beta}$, we obtain,

$$\begin{aligned} \mathrm{E}\left[\hat{\beta}_{\mathrm{Ridge}} \mid \mathbf{X}\right] &= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I} - \ \lambda\mathbf{I}\right)\boldsymbol{\beta} \\ &= \left[\mathbf{I} - \ \lambda\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\ \right]\boldsymbol{\beta} \\ &= \boldsymbol{\beta} - \ \lambda\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\beta}. \end{aligned} \tag{3.30}$$

This means that the bias of the ridge estimator is proportional to λ. That is, the larger is λ, the larger is the bias of the ridge estimator with respect to $\boldsymbol{\beta}$. Hence, this ridge estimator accepts a little bias to reduce the variance and the mean squared error, respectively of the estimates and may improves the prediction accuracy.

Although the ridge estimator has a greater bias, it possesses a smaller variance than OLS estimator. To show that we want to find the total variances of two methods by taking trace of the variance matrices. For the OLS, using the SVD decomposition presented earlier, we have

$$\operatorname{tr}\left(\operatorname{Var}\left[\hat{\boldsymbol{\beta}}_{\mathrm{OLS}} \mid \mathbf{X}\right]\right) = \operatorname{tr}\left(\sigma^2 \left(\mathbf{X}'\mathbf{X}\right)^{-1}\right)$$

$$= \sigma^2 \sum_{\mathrm{j}=1}^{\mathrm{p}} \frac{1}{d_{\mathrm{j}}^2} \tag{3.31}$$

but for the ridge estimator, we obtain

$$\operatorname{Var}\left[\hat{\beta}_{\mathrm{Ridge}} \mid \mathbf{X}\right] = \sigma^2 \left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \left(\mathbf{X}'\mathbf{X}\right)\left(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \tag{3.32}$$

By using the SVD decomposition, it follows

$$\operatorname{Var}\left[\hat{\beta}_{\mathrm{Ridge}} \mid \mathbf{X}\right] = \sigma^2\, \mathbf{V}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1} \mathbf{V}'\mathbf{V}\mathbf{D}^2\mathbf{V}'\mathbf{V}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1} \mathbf{V}', \tag{3.33}$$

As before, since $\mathbf{X}'\mathbf{X}$ and $\lambda\mathbf{I}$ are simultaneously diagonalizable, then we have,

$$\left(\mathbf{V}\mathbf{D}^2\mathbf{V}' + \lambda\mathbf{I}\right)^{-1} = \mathbf{V}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1} \mathbf{V}'. \tag{3.34}$$

which simplifies to

$$\operatorname{Var}\left[\hat{\beta}_{\mathrm{Ridge}} \mid \mathbf{X}\right] = \sigma^2\, \mathbf{V}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1} \mathbf{D}^2 \left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1} \mathbf{V}' \tag{3.35}$$

by taking the trace of this quantity as the sum of the eigenvalues, then it follows that

$$\operatorname{tr}\left(\operatorname{Var}\left[\hat{\boldsymbol{\beta}}_{\mathrm{Ridge}} \mid \mathbf{X}\right]\right) = \sigma^2 \sum_{\mathrm{j}=1}^{\mathrm{p}} \frac{d_{\mathrm{j}}^2}{\left(d_{\mathrm{j}}^2 + \lambda\right)^2}\,. \tag{3.36}$$

It can be show after some algebraic simplification that

$$\operatorname{tr}\left(\operatorname{Var}\left[\hat{\boldsymbol{\beta}}_{\mathrm{OLS}} \mid \mathbf{X}\right]\right) \geq \operatorname{tr}\left(\operatorname{Var}\left[\hat{\boldsymbol{\beta}}_{\mathrm{Ridge}} \mid \mathbf{X}\right]\right). \tag{3.37}$$

3.2.3 Ridge coefficient path

Although all p coefficients in a ridge fit will be non-zero, they are fitted in a restricted fashion controlled by λ.

Figure 3.1 shows the ridge coefficient estimates for the prostate cancer example taken from [58], plotted as functions of df(λ), the effective degrees of freedom implied by the penalty λ.

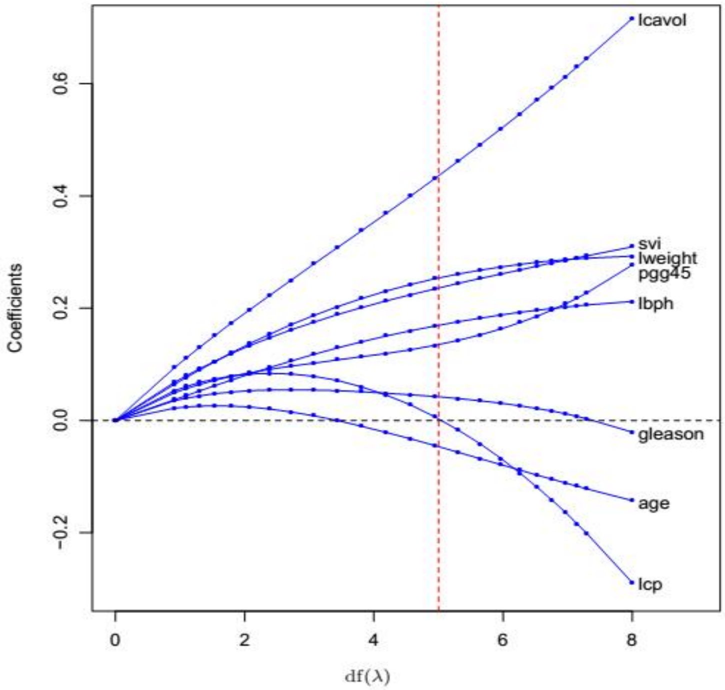

Figure 3.1 Ridge coefficients path for the prostate cancer example, as the tuning parameter λ is varied. Coefficients are plotted versus df(λ), the effective degrees of freedom. A vertical line is drawn at df = 5.0, the value chosen by cross-validation [58].

In summary, Ridge regression achieves a stable fit even in the presence of strongly correlated predictors, shrinking each coefficient based on the variation of the corresponding variable. However, penalty parameter does not force coefficients of poor variables to exactly zero, so ridge regression does not select predictors and therefore does not give an easily interpretable model. For this reason, Tibshirani in his seminal paper [2] developed the absolute shrinkage and selection operator (Lasso).

3.3 Absolute shrinkage and selection operator (Lasso)

A popular penalized least squares method that does both continuous shrinkage and automatic variable selection simultaneously is the absolute shrinkage and selection operator (Lasso), proposed by Tibshirani [2]. Hence, it is suitable to estimate coefficients and perform variable selection for high dimensional data.

The lasso estimates are obtained by minimizing the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. The Lasso estimate is defined by

$$\hat{\boldsymbol{\beta}}_{Lasso} = \underset{\beta \in \Re^p}{\arg\min} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i' \boldsymbol{\beta} \right)^2 \ , \quad s.t. \ \sum_{\mathrm{j}=1}^{\mathrm{p}} \left| \beta_{\mathrm{j}} \right| \leq \mathrm{t} \ , \ \mathrm{t} \geq 0 \tag{3.38}$$

Thus, Lasso estimator substitutes the L_2-norm of the ridge estimator and imposes the L_1-norm on the regression coefficient. Like ridge regression, it can be written in lagrangian form as:

$$\hat{\boldsymbol{\beta}}_{Lasso} = \underset{\beta \in \Re^p}{\arg\min} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i' \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \left| \beta_j \right| \ , \ \lambda \geq 0 \tag{3.39}$$

or,

$$\hat{\boldsymbol{\beta}}_{Lasso} = \underset{\beta \in \Re^p}{\arg\min} = \left\| \boldsymbol{y} - \mathbf{X}\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \tag{3.40}$$

where the regularization parameter λ has a one-to-one correspondence with the parameter t of Equation (3.38).

Because of the nature of Lasso constraint, the Lasso does a kind of continuous subset selection. If values of the parameter t is chosen to be sufficiently small, the estimated Lasso coefficients are shrunk towards zero and some coefficients are exactly set to zero; for t = 0 all of them are equal to zero. If values of t is chosen such that $t < t_0$ with t_0 = $\sum_{j=1}^{p} \left| \hat{\beta}_{j,oLs} \right|$ this cause a shrinkage of the coefficients, for example, when t = t_0/2, the least squares coefficients are shrunk by about 50% on average. Whereas choosing of t > t_0 making the estimated Lasso coefficients similar of least squares $\hat{\beta}_{j,oLs}$.

Figure 3.2 shows the optimization problem for two variables with various penalties showing subset selection property of Lasso. The residual sum of squares has elliptical contours. The constraint region for the lasso is the diamond $\left|\beta_1\right| + \left|\beta_2\right| \leq t$, whereas that for ridge regression is the disk $\beta_1^2 + \beta_2^2 \leq t$. The solution of the optimization problem is given at the point where the elliptical contour of the least squares function hits the constraint region. The diamond has corners; if the solution occurs at a corner, then it has one parameter β_j equal to zero. In contrast, there are no vertices in the disk that can be touched; ridge regression cannot produce zero solutions.

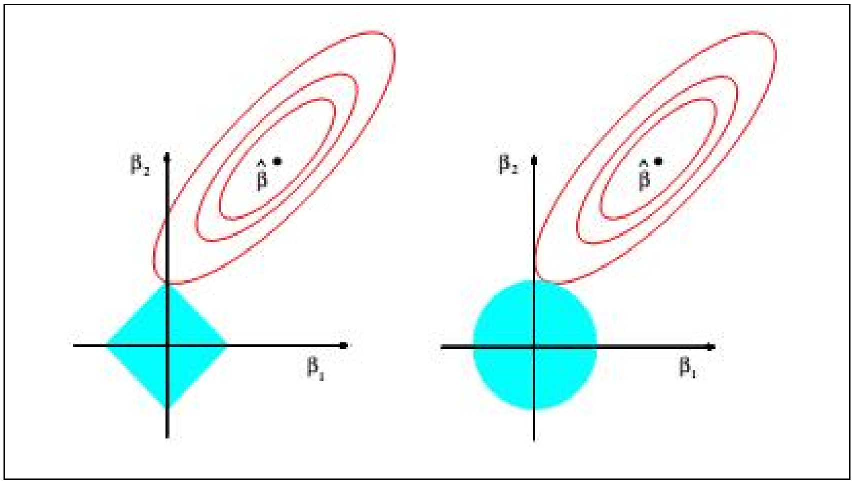

Figure 3.2 The optimization problem of Lasso (left) and ridge regression (right) for two variables with various penalties. Shown are contours of the least squares and constraint functions. The solid blue areas are the constraint regions $|\beta_1|+|\beta_2|\le t$ and $\beta_1^2+\beta_2^2\le t$, respectively, while the red ellipses are the contours of the least squares function [58].

When p > 2, the diamond becomes a rhomboid, and has many corners, flat edges and faces; there are many more opportunities for the estimated parameters to be zero.

Unlike ridge regression, Lasso imposes the L$_1$-penalty on the regression coefficients which is not differentiable at zero, therefore there is no closed form expression as in ridge regression. However, in the case of an orthonormal input matrix X, the lasso solution has closed-form as

$$\hat{\beta}_{j,Lasso} = sign(\hat{\beta}_{j,oLS})(\hat{\beta}_{j,oLS} - \lambda)_+, \quad j \in \{1,\dots,p\}, \tag{3.41}$$

where sign denotes the sign of its argument (±1), and $(.)_+$ indicates the positive part. This solution represents so-called soft-thresholding because Lasso translates each coefficient by a constant factor λ, truncating at zero, whereas Best-subset selection method drops all variables with coefficients smaller than the Mth largest; this is a form of "hard-thresholding." In other hand, ridge regression does a proportional shrinkage.

Table 3.1. presents the closed solutions of three procedures where input matrix **X** is an orthonormal. Each method applies a simple transformation to the least squares estimate $\hat{\beta}_j$. In addition, figure 3.3 the compares three procedures of a regression coefficient in the orthonormal case. Estimations shown by broken red lines. The **x**-axis represents the unrestricted coefficient $\hat{\beta}_{\mathrm{j,OLS}}$, and the **y**-axis represents the corresponding regularized coefficient $\hat{\beta}_{\mathrm{j}}$. The 45◦ line in gray shows in each panel correspond to the unrestricted least squares estimation

Table 3.1 Estimators of β_j in the case of orthonormal columns of **X**. M and λ are constants chosen by the corresponding techniques.

Estimator	**Formula**
Best-subset (size *M*)	$\hat{\beta}_{\mathrm{j,OLS}} \cdot I(\hat{\beta}_{\mathrm{j,OLS}} \geq \hat{\beta}_{(M)})$
Ridge	$\dfrac{\hat{\beta}_{\mathrm{j,OLS}}}{1 + \lambda}$
Lasso	$sign(\hat{\beta}_{j,OLS})(\hat{\beta}_{j,OLS} - \lambda)_+$

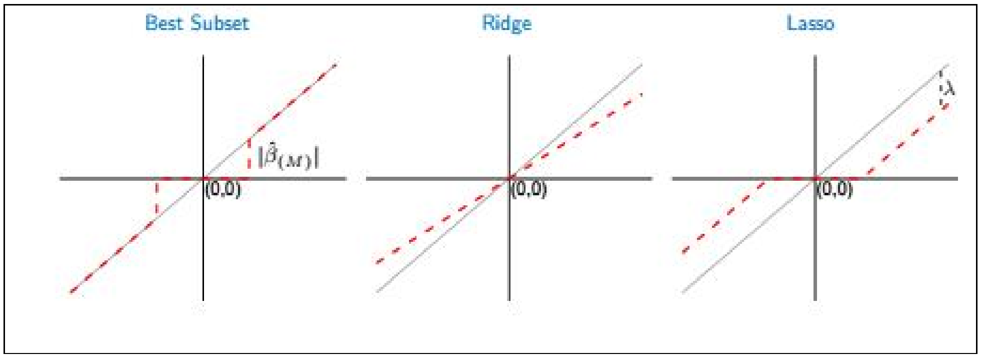

Figure 3.3 Best-subset, Lasso, Ridge estimations of a regression coefficient in the orthonormal case. The 45◦ line in gray line corresponds to the unrestricted least squares estimation [58].

3.3.1 Degrees of freedom of the Lasso

The concept of degrees of freedom plays an important role in quantifying the model complexity [58]. So it is of great interest to know the degrees of freedom of the lasso for any given regularization parameter λ for selecting the optimal lasso model.

Many model selection criteria can be used to get the optimal hyperparameters of the Lasso estimator in the sense of the prediction, e.g. C_p, AIC, BIC and GCV (Generalized Cross Validation) and the degrees of freedom is obtained after penalizing the fitted model.

There are different ways to define degree of freedom. Stein's unbiased risk estimation (SURE) theory [58] gives a rigorous definition of the degrees of freedom for any fitting procedure in a classical linear model as in (2.7) with fixed design and errors $\varepsilon_i \sim N(0, \sigma^2)$. Let **Hy** = $\hat{\mathbf{y}}$ the hat-operator which maps the response vector $\mathbf{y} = (y_1, \ldots, y_n)'$ to its fitted values $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)'$. It is shown by [60] that the degrees of freedom for a possibly non-linear hat-operator **H** are then defined as

$$\mathrm{df}(\mathbf{H}) = \sum_{i=1}^{n} Cov\,(\hat{y}_i, y_i) / \sigma^2, \tag{3.42}$$

where the values $\hat{y}_i$ arise from any model fitting method.

In case of parameter estimation with the maximum-likelihood method, the degrees of freedom equal the number of parameters we need to estimate. However, generally, there is no exact relation between the degrees of freedom and the number of parameters in the model [61]). For example, fitted model may has one parameter, but the degrees of freedom is greater than one. Alternatively, the standard formula for degrees of freedom of linear hat-operators $\mathbf{Hy} = \hat{\mathbf{y}}$ with a hat-matrix **H**, can be used in a classical linear model with independent errors [62]; since $Cov\,(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sigma^2\mathbf{H}$, the number of degrees of freedom is given by the trace of the linear hat operators, df(**H**) = tr(**H**).

Unfortunately, because the lasso is a nonlinear fitting method, and an analytical expression is hard to derive except special case of an orthonormal design, this formula cannot be applied. Thus it is difficult to derive the analytical expression of the degrees of freedom of Lasso, except for the low dimensional case where $p \leq n$.

To overcome the analytical difficulty, [61, 63] proposed using a data perturbation technique to numerically compute an (approximately) unbiased estimate for df($\mathbf{H}_\lambda$) when the analytical form of $\hat{\mathbf{y}}$ is unavailable. The bootstrapping techniques [64] can also be used to obtain an (approximately) unbiased estimator of the degrees of freedom. However, this approach can be computationally expensive.

Another approach to count the number of degrees of freedom is by the number of non-zero estimated parameters. Let the selected model, for a given λ, given as

$$\hat{S}(\lambda) = \{j\,; \hat{\beta}_j \neq 0, j = 1, \,.\,.\,.\,, p\}. \tag{3.43}$$

now for the low dimensional case where $p \leq n$. The degrees of freedom can be estimating by the number of non-zero estimated parameters, i.e. $|\hat{S}\,|$. It may seem that by shrinking the coefficients, the number of degrees of freedom is reduced, but this is compensated by the lasso's "freedom" for selecting certain variables and discarding others.

However, for the Lasso with penalty parameter λ and associated hat-operator **H**= **H**(λ), and if rank(X) = p (low dimensional case), the degrees of freedom of the lasso estimator is estimated by the expected number of selected variables from the Lasso estimator [61]

$\mathrm{df}(\mathbf{H}) = \mathrm{E}[|\hat{S}|]$, (3.44)

[65] proposed that the number of nonzero predictors in the model is an unbiased estimate for the degrees of freedom.

$\widehat{df}(\mathbf{H})=|\hat{S}|$. (3.45)

Unfortunately, this simple formula has been proved only for the $p < n$ case and it is not known if the result holds when **X** is not full-rank [5].

According to information above the regularization parameter λ can be chosen by the BIC criterion as

$$\hat{\lambda}_{\mathrm{BIC}} = \mathrm{argmin}_{\lambda}\,(n \ \log(n^{-1}\left\|\mathbf{y} - \mathbf{H}(\lambda)\mathbf{y}\right\|^{2}) + \log(n) \cdot |\hat{S}(\lambda)|). \tag{3.46}$$

3.3.2 Entire regularization path

The estimation of Lasso is a convex optimization problem and can be solved by a quadratic programming algorithm for a given λ. This can be computationally expensive, since it requires computing the estimator $\hat{\beta}(\lambda)$ for a grid of λs. For example, selection of a good value λ, e.g., by using cross-validation, requires the computation over many different candidate values. However, it is possible to compute the entire regularized optimal path over all values of λ as the following sense. For linear models, the regularized optimal path $\{\hat{\beta}(\lambda); \lambda \in \Re^{+}\}$ is piecewise linear as a function of λ [66].

$\exists \lambda_0 = 0 < \lambda_1 < \lambda_{m-1} < \lambda_m = \infty,\ \gamma_0,\ \gamma_1, \ldots,\ \gamma_{m-1} \in \Re^{p}$ such that

$$\hat{\beta}(\lambda) = \hat{\beta}(\lambda_k) + (\lambda - \lambda_k)\gamma_k \quad \text{for } \lambda_k \le \lambda < \lambda_{k+1}\ (0 \le k \le m - 1). \tag{3.47}$$

There is a maximal value $\lambda_{\max} = \lambda_{m-1}$ where $\hat{\beta}(\lambda) = 0$ for all $\lambda \ge \lambda_{\max}$ and $\hat{\beta}_j(\lambda) \neq 0$ for $\lambda < \lambda_{\max}$ and some j. The value $\lambda_{\max}$ can defined as:

$$\lambda_{max} = \max_{1 \le j \le p} |2\mathbf{x}_j \boldsymbol{y}|/n. \tag{3.48}$$

Typically, every λ_k is a kink point (correspond to changes in slope of the estimated trend) for only a single component of the coefficient paths $\hat{\beta}(.)$. The number of different λ_k-values is of the order $m = O(n)$ [66].

Figure 3.4 taken from [67] shows the entire regularization path for Lasso in a linear model, based on a real data example with $n = 71$ and $p = 4088$.

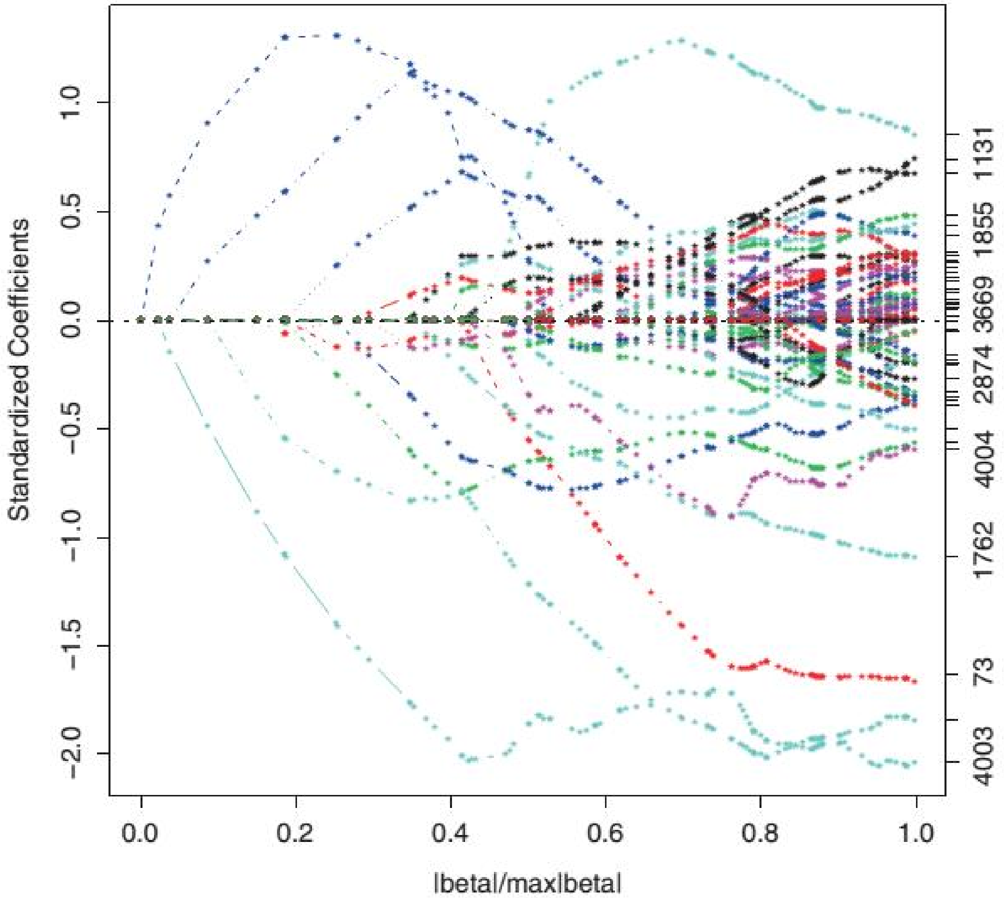

Figure 3.4 Regularization path for Lasso in a linear model with n = 71 and p = 4088. x-axis: $\left\|\hat{\beta}(\lambda)\right\|_1 \Big/ \max\left\{\left\|\hat{\beta}(\lambda)\right\|_1 ; \lambda\right\}$, and y-axis: $\hat{\beta}\sqrt{\hat{\sigma}_j^2\ (n\ -1)}$ where $\hat{\sigma}_j^2$ denotes the empirical variance of $\mathbf{x}_j$ [67] .

Thus, if the values (λ_k, γ_k) ($k = 0, \ldots, m - 1$) are computed, the entire regularized optimal path can be easily generated by sequentially calculating the "step sizes" between each two consecutive λ_k values and the "directions" $\gamma_1,\ldots,\gamma_{m-1}$.

There are efficient algorithms for computing the entire regularized optimal path as λ is varied, such as the least angle regression (LARS) algorithm, introduced by [6]. LARS has been modified to compute entire regularization path of Lasso with small computational cost. Specifically, because the regularization path is piecewise linear for the lasso, the solution is computed for only finite number of λ values. The computational complexity of (LARS) algorithm for computing the entire regularization path is:

$O(np \min(n , p))$ essential operation counts. (3.49)

Hence, if $p > n$, $O(np \min(n , p)) = O(p)$ computational complexity which is linear in the dimensionality p.

However, the non-squared losses functions can be used in a wider range of optimization problems. For example, the loss of function may be differentiable and convex and the penalty is convex and separable [68]. Also, penalty function may has a group structure as we will see in subsection (3.3.4), In this situations, entire path-following algorithms such as the LARS-algorithm are not available and coordinate descent algorithms can deal with these optimization problems.

Although LARS is considerably fast, pathwise coordinate descent optimization algorithms can be more efficient than LARS in high-dimensional settings [69, 70, 71]. suggest coordinate descent for solving convex statistical problems such as the lasso. For inexplicable reasons, they did not follow up their theoretical suggestions with numerical confirmation for highly underdetermined problems. The theory under pathwise coordinate optimization is developed by [72, 73].

3.3.3 Asymptotic Properties of Lasso

Statistical performance of Lasso can be measured in two different ways: best prediction accuracy and how well does the lasso estimate the true regression coefficients. In the second case, assuming that the true parameter vector is indeed sparse, recovering the true sparsity pattern is often a primary goal. Therefore, it is important to study asymptotic properties: model selection consistency and prediction consistency.

3.3.3.1 The asymptotic properties when p is fixed

The asymptotic properties have been extensively studied and analyzed. Under some regularity conditions on the design, [3] first derived the asymptotic distribution of the Lasso estimator and proved its estimation consistency for Lasso for fixed p and fixed $\boldsymbol{\beta}$ (i.e., p and $\boldsymbol{\beta}$ are independent of n) as n → ∞. In particular, they have shown that under the shrinkage rate $\lambda_n = o(\sqrt{n})$ and $\lambda_n = o(n)$, Lasso has estimation consistency propriety, i.e. $\hat{\boldsymbol{\beta}}(\lambda) \rightarrow_p \boldsymbol{\beta}$. Moreover if errors terms are iid and has a common finite second moment σ^2, the $\sqrt{n}$ scaled Lasso estimator with $\{\lambda_n\}_n \in$ N has an asymptotic normal distribution with variance $\sigma^2\,\boldsymbol{\Sigma}^{-1}$, where $\hat{\boldsymbol{\Sigma}} = n^{-1}\mathbf{X}_n'\mathbf{X}_n, \rightarrow \boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}$ is a positive definite matrix. On the other hand, [74] have shown that for a fixed p and orthogonal designs, the Lasso estimate does not give consistent model selection with the optimal prediction criterion such as cross-validation (CV).

3.3.3.2 The asymptotic properties for high dimensions

This subsection discuses the asymptotic results in [67]. The asymptotic is considered with respect to a triangular array of observations:

Assume that p = $\mathrm{p_n}$, X = $\mathrm{X_n}$ = $(x^{\mathrm{j}}_{\mathrm{n;i}})$, and $\boldsymbol{\beta}=\boldsymbol{\beta}_{\mathrm{n}}=(\beta_{\mathrm{n;j}})$ all changes as $\mathrm{n}\to\infty$, typically with $\mathrm{p}\to\infty$ faster than n i.e. with $\mathrm{p_n/n}\to\infty$ as $\mathrm{n}\to\infty$. Thus, one considers a sequence of models

$$\mathrm{y_{n;i}}=\sum_{\mathrm{j=1}}^{\mathrm{p_n}}\beta^{0}_{\mathrm{n;j}}x^{\mathrm{j}}_{\mathrm{n;i}}+\varepsilon_{\mathrm{n;i}},\quad \mathrm{i}=1,\,\ldots,n;\; n=1,2,\,\ldots \tag{3.50}$$

The assumptions about $\varepsilon_{n;i}$ are as in the linear model (2.7). A prediction consistency result requires a sparsity assumption of the form

$$\left\|\boldsymbol{\beta}^{0}\right\|_{1}=\left\|\boldsymbol{\beta}^{0}_{\mathrm{n}}\right\|_{1}=o\left(\sqrt{\frac{n}{\log(p_{\mathrm{n}})}}\right), \tag{3.51}$$

where $\boldsymbol{\beta}^{0}$ is the true parameter vector. Under some regularity conditions on the error distribution, for a suitable range of $\lambda=\lambda_n\asymp\sqrt{log\left(p\right)/n}$,

the Lasso is consistent for estimating the underlying regression function:

$$\left(\hat{\boldsymbol{\beta}}(\lambda)-\boldsymbol{\beta}^{0}\right)'\Sigma_{X}\left(\hat{\boldsymbol{\beta}}(\lambda)-\boldsymbol{\beta}^{0}\right)=o_{P}(1)\;\left(n\;\to\infty\right), \tag{3.52}$$

where $\boldsymbol{\Sigma}_{X}$ equals $n^{-1}\mathbf{X}'\mathbf{X}$ in case of a fixed design, and equal to the covariance matrix of the covariates vector **X**, for a random design, and $Z_n=o_p(1)$ means that $P(|Z_n|>\epsilon)\to\infty$ as $n\;\to\infty$, for any ϵ > 0, i.e. that Z_n tends to zero in probability).

The left hand side in quantity in (3.52) can be written as the average squared error loss:

$$\left\|\mathbf{X}(\hat{\boldsymbol{\beta}}(\lambda)-\boldsymbol{\beta}^{0})\right\|_{2}^{2}\Big/n\;\text{ for fixed design}, \tag{3.53}$$

$$\mathrm{E}[(X_{new}\,(\hat{\boldsymbol{\beta}}(\lambda)-\boldsymbol{\beta}^{0}))^{2}]\;\text{ for random design} \tag{3.54}$$

where E is taken with respect to the new test observation X_{new} (a 1 × p vector) .

In general, an oracle inequality compares the performance of an estimator to some "best estimator", the so-called oracle, which relies on perfect information and which is not available in practice. Now for λ in a suitable range of the order $\sqrt{log\left(p\right)/n}$, and under certain compatibility conditions on the design **X**, the so-called oracle inequality for fixed design can be shown as

$$\mathrm{E}\left[\left\|\mathbf{X}(\hat{\beta}(\lambda)-\beta^0)\right\|_2^2\Big/n\right]=O\left(\frac{|S_0|\log(p)}{n\varphi^2}\right), \tag{3.55}$$

where $|S_0|$ = card(S_0) and S_0 = $\left\{j\,;\beta_j^0\neq\ 0\right\}$ is the active set of variables, which contains all covariates with non-zero corresponding regression coefficients, and φ^2 is the so-called compatibility constant or restricted eigenvalue which is a number depending on the compatibility between the design and the L_1 -norm of the regression coefficient.

The "oracle inequality" at best is bounded below by a positive constant by choosing λ (of order $\sigma\log(p)$), [75, 76]

$$\left\|\mathbf{X}(\hat{\boldsymbol{\beta}}(\lambda)-\boldsymbol{\beta}^0)\right\|_2^2\Big/n\leq 4\big/\varphi^2\log(p)\frac{\sigma^2}{n}|S_0|, \tag{3.56}$$

From standard least squares theory, we know that

$$\mathrm{E}\left[\left\|\mathbf{X}(\hat{\boldsymbol{\beta}}_{OLS}-\boldsymbol{\beta}^0)\right\|_2^2\Big/n\right]=\frac{\sigma^2}{n}p. \tag{3.57}$$

Note that, up to the log(p)-term (and the compatibility constant φ^2), the mean-squared prediction error in the Lasso estimator 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 $|S_0|$ variables only. Because we do not know the active set S_0 m additional (log p)-factor is added to the right hand side of inequality (3.56) [77].

Estimation consistency

Estimation consistency can be obtained under similar conditions as for prediction consistency.

$$\left\|\left(\hat{\beta}(\lambda)-\beta^0\right)\right\|_q=o_p(1)\text{ , for q =1 or 2} \tag{3.58}$$

where $\left\|\beta\right\|_q=\left(\sum_j|\beta j\,|^q\right)^{1/q}$.

Variable screening

In a setting as (3.50), consider the active set of variables S_0 is allowed to depend on n i.e.

$$S_0 = S_{0;n} = \left\{ j\, ; \beta^0_{n;j} \neq 0, j = 1, \ldots, p_n \right\} \qquad (3.59)$$

Since the Lasso estimator is selecting some variables, this means some of the coefficients are exactly zero, in this case Lasso estimator is considered as screening estimator:

$$\hat{S}_{S_{0;n}}(\lambda) = \{ j\, ; \hat{\beta}_j(\lambda) \neq 0, j = 1, \ldots, p \}. \qquad (3.60)$$

The following lemma (Lemma 3.1 [67]) proved the uniqueness $\hat{S}_{S_{0;n}}(\lambda)$ across different solutions $\hat{\beta}_j(\lambda)$ of the optimization in (3.39).

Lemma 3.1. Let the gradient of $n^{-1} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2$ by $G(\beta) = 2\mathbf{X}'(\mathbf{Y} - \mathbf{X}\beta)/n$. Then a necessary and sufficient condition for $\hat{\beta}$ to be a solution of (3.39) is:

$$G_j\left(\hat{\beta}\right) = -sign\left(\hat{\beta}_j\right)\lambda \text{ if } \hat{\beta}_j \neq 0. \qquad (3.61)$$

$$\left|G_j\left(\hat{\beta}\right)\right| \leq \lambda \text{ if } \hat{\beta}_j = 0. \qquad (3.62)$$

Typically, the Lasso solutions $\beta_{n;j}$ are not unique (e.g. they are not unique if p > n but from Lemma 3.1 still $\hat{S}_{S_{0;n}}$ is unique. Further, $\left|\hat{S}_{S_{0;n}}\right| \leq$ min (n, p_n).

In practice, the goal is to find at least the covariates with significant absolute values of the regression coefficients$|\beta_j|$. In other formal words, we want to find the true set of ”relevant” variables

$$S_0^{relevant}\left(C\right) = \{ j ; |\beta^0_j| \geq C, j = 1, \ldots, p \}, \text{ for some } C > 0. \qquad (3.63)$$

Using the result in (3.58), which holds under weaker assumptions than the restrictive neighborhood stability or irrepresentable condition: for any fixed 0 < C < ∞, it can be shown that:

$$\mathbf{P}\left[\hat{S}(\lambda) \supset S_0^{relevant(C)}\right] \to 1 \; (n \to \infty). \qquad (3.64)$$

This means that under suitable conditions, $\hat{S}_{0;n}(\lambda)$ will contain $S_{0;n}^{relevant\,(C)}$ asymptotically, but often, depending on the value of λ, it will contain many more variables.

In general, let

$$\left\| \hat{\beta}_n(\lambda_n) - \beta^0 \right\|_1 \leq a_n \text{ with high probability,} \tag{3.65}$$

under compatibility conditions on the design matrix **X**, with λ_n in the range of order $\sqrt{log\,(p_n)/n}$, it holds that $a_n = O\left(|S_0|\sqrt{log\,(p_n)/n}\right)$. Then, for $C_n > a_n$:

$$\mathbf{P}\left[\hat{S}_n(\lambda_n) \supset S_0^{relevant(C_n)}\right] \rightarrow 1\ (p \geq n \rightarrow \infty) \tag{3.66}$$

when assuming the beta-min condition

$$\min_{j \in S_0^C} |\beta_j^0| \gg \varphi^{-2}\sqrt{|S_0|\,log\,(p)/n}\,, \tag{3.67}$$

then all non-zero coefficients are at least as large as C_n in absolute value. In this case $S_0^{relevant(C_n)}$ may equal to S_0, and then

$$\mathbf{P}\left[\hat{S}_n(\lambda_n) \supset S_0\right] \rightarrow 1\ (p \geq n \rightarrow \infty) \tag{3.68}$$

The properties in (3.64) and (3.66) are called variable screening. Regarding to the subsection (3.3.2) in the analysis of the LARS algorithm, Lasso estimated model has relevant covariates smaller or equal to min (n, p) if $p \gg n$, min $(n, p) = n$: hence, Lasso estimator makes a large dimensionality reduction in terms of the original covariates.

Variable selection

Consider the selected model, for a given λ, using the Lasso as in (3.43), then all possible Lasso sub-models can be computed as

$$\hat{\mathscr{S}} = \{\hat{S}(\lambda);\ all\,\lambda\} \tag{3.69}$$

with $O(np\ \min(n, p))$ operation counts. As pointed out above, every sub-model in $\hat{\mathscr{S}}$ cardinality smaller or equal to min (n, p). Furthermore, the number of sub-models in $\hat{\mathscr{S}}$ is typically of the order O (min (n, p)) [66]. Thus, in summary, each Lasso estimated sub-model contains at most min (n, p) variables.

$|\hat{S}(\lambda)| \leq \min(n, p)$ for every λ, which is a small number if $p \gg n$, and the number of different Lasso estimated sub-models is typically $|\hat{\mathscr{S}}| = O(\min(n, p))$,

which represents a huge reduction compared to all 2^p possible sub-models if $p \gg n$.

Assuming rather restrictive conditions, that with probability tending to 1, $S_0 \in \hat{\mathscr{S}}$ that the Lasso is appropriate for addressing the problem of variable selection.

The condition is so-called neighborhood stability or irrepresentable condition. Under such a neighborhood stability condition, and assuming that the non-zero regression coefficients satisfy

$$\dot{I}\mathrm{nf}_{j \in S_0^C} \, |\beta_j^0| \gg \sqrt{|S_0| \, log(p)/n} \, , \tag{3.70}$$

([78]. Theorems 1 and 2) show the following: for a suitable $\lambda = \lambda_n \gg \sqrt{log(p)/n}$,

$$\mathbf{P}\left[\hat{S}(\lambda) = S_0^{relevant(C)}\right] \rightarrow 1 \; (n \rightarrow \infty). \tag{3.71}$$

Neighborhood stability and irrepresentable condition

[78] have introduced the so-called neighborhood stability condition for consistent variable selection with the Lasso. The so-called irrepresentable condition, introduced by [65] is another condition on the design matrix equivalent to the neighborhood stability condition n (at least for the case where $n > p$ is fixed) and easier to describe.

Definition 3.1: Strong Irrepresentable Condition. Let covariance of the predictor variables (the Gram matrix) $\hat{\Sigma}$ equals $n^{-1}\mathbf{X}'\mathbf{X}$, and assume the active set $S_0 = \left\{ j \, ; \beta_{n;j}^0 \neq 0 \right\}$ $= \{1 , . . . , |S_0|\}$ consists of the first $|S_0|$ variables. Let

$\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_{1,1} & \hat{\Sigma}_{1,2} \\ \hat{\Sigma}_{2,1} & \hat{\Sigma}_{2,2} \end{pmatrix}$, where $\hat{\Sigma}_{1,1}$ is a $|S_0| \times |S_0|$ matrix corresponding to the active variables, $\hat{\Sigma}_{1,2} = \hat{\Sigma}_{2,1}$ is a $|S_0| \times (p - |S_0|)$ matrix and $\hat{\Sigma}_{2,2}$ a $(p - |S_0|) \times (p - |S_0|)$ matrix. Then the strong irrepresentable condition defined as:

$$\left\| \hat{\Sigma}_{2,1} \hat{\Sigma}_{1,1}^{-1} sign\left(\beta_1^0, \, . \, . \, . \, , \, \beta_{|S_0|}^0\right) \right\|_\infty \leq \theta \text{ for some } 0 < \theta < 1, \tag{3.72}$$

and weak Irrepresentable Condition defined as:

$$\left\| \hat{\Sigma}_{2,1} \hat{\Sigma}_{1,1}^{-1} sign\left(\beta_1^0, \, . \, . \, . \, , \, \beta_{|S_0|}^0\right) \right\|_\infty \leq 1 \tag{3.73}$$

where $\|x\|_\infty = max_j \, |x^{(j)}|$ and $sign\left(\beta_1^0, \ldots, \beta_p^0\right) = \left(sign\left(\beta_1^0\right), \ldots, sign\left(\beta_p^0\right)\right)'$.

As with the neighborhood stability condition, [4] found the irrepresentable condition in (3.72) is a sufficient and necessary condition on the design matrix for the Lasso to be consistent variable selection (model selection). [4, 78] have shown that under a strong irrepresentable condition, Lasso is consistent for variable selection even when the number of variables p grows faster than n even p is $\exp(n^a)$ for some $0 < a < 1$.

In summary, the Lasso is variable selection consistent under certain conditions, but not in general. The restrictive conditions on the design have some relevant implications on the statistical practice for high-dimensional model selection: with strongly correlated design, the Lasso is not efficient for estimating the nonzero parameters.

3.3.4 Elastic net

Although Lasso does both variable selection and continuous shrinkage simultaneously, there are some limitations making the lasso an inappropriate variable selection method:

1. In the case $p > n$, and regarding to optimization problem of Lasso, the estimated model of Lasso has relevant covariates smaller or equal to min (n, p) = n (i.e. it selects at most n variables before it saturates. This means that a limiting feature for a variable selection method.
2. If is a group of predictors with high pairwise correlations among them, the Lasso tends to drop all but one from this group and does not group predictors as pointed out by [79].
3. In the $n > p$ case, if predictors are high correlated, the prediction performance of the lasso is dominated by ridge regression [2].

For example, in the analysis on microarray dataset which has several thousands of predictors (genes) and often a small number of samples, there is a goal to identify all the genes involved in a particular process, although their expression levels are very similar. This kind is a grouped variables situation, in which the Lasso is not capable of grouped variable selection, because it can only select at most n variables out of p candidates [64], and it lacks the ability to detect the grouping information.

The elastic net, proposed by [79], is a regularization technique that does automatic variable selection and continuous shrinkage as lasso, and is capable of grouped variable selections when groups of predictors are highly correlated as ridge.

Mathematically, the elastic net criterion can be defined as:

$$\hat{\beta}_{elastic\ net} = \underset{\beta\in\Re^p}{\arg\min}\left\{\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2 + \lambda_1\sum_{j=1}^{p}\left|\beta_j\right| + \lambda_2\sum_{j=1}^{p}\beta_j^2\right\},\ \lambda_1,\lambda_2 \ge 0 \qquad (3.74)$$

where λ_1 and λ_2 are the penalties.

Let $\alpha^* = \dfrac{\lambda_2}{\lambda_1+\lambda_2}$, then solving $\hat{\beta}$ in (3.74) is equivalent to the optimization problem:

$$\hat{\beta}_{elastic\ net} = \underset{\beta\in\Re^p}{\arg\min}\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2,\quad s.t.\ (1\text{-}\alpha^*)\sum_{j=1}^{p}\left|\beta_j\right| + \alpha^*\sum_{j=1}^{p}\beta_j^2 \le \text{t}. \qquad (3.75)$$

where the function $(1\text{-}\alpha)\sum_{j=1}^{p}\left|\beta_j\right| + \alpha\sum_{j=1}^{p}\beta_j^2$ is the elastic net penalty, which is a convex combination of the lasso and ridge penalty. ∀ α^* ∈ (0, 1), the elastic net penalty function has no first derivative) at 0 and it is strictly convex. Figure 3.5 shows the contour plot of the elastic net penalty in 2-dimensional space.

Equation (3.75) is so-called the naive elastic net, because it is similar to either ridge regression or the lasso and tends to over shrink in regression problems. For α^* = 1, the naive elastic net problem is equivalent to ridge problem, whereas for α^* = 0, it is equivalent to lasso problem.

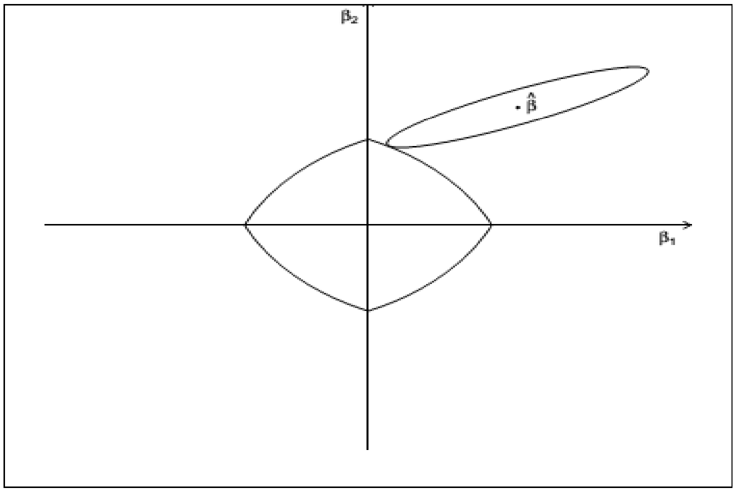

Figure 3.5 The curve around origin is the contour plot of the elastic net penalty with p = 2, α^* = 1 and t = 0.5.

CHAPTER 4

ROBUST REGULARIZED REGRESSION METHODS

4.1 LAD-Lasso

Recall that the OLS estimate is not robust to outliers or heavy-tailed errors in a response variable. Since the Lasso is penalized least squares, it suffers from unusual data points like the OLS estimate. To achieve further robustness, [18] proposed a LAD-Lasso that uses the sum of the absolute values of the residuals instead of sum of squares. Thus, LAD-Lasso can do regression shrinkage and selection like Lasso and is resistant to outliers or heavy-tailed errors like LAD.

$$\hat{\boldsymbol{\beta}}_{LAD-lasso} = \arg\min_{\beta\in\Re^{p}} \; L(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left|y_i - x_i'\boldsymbol{\beta}\right| + \lambda\sum_{j=1}^{p}|\boldsymbol{\beta}_j| \tag{4.1}$$

4.1.1 Computation of the LAD-Lasso

The following procedure is to compute LAD-Lasso as proposed by [18]. For any given λ_n, let $\{(y_i^*, x_{i1}^*, \cdots x_{ip}^*)\}, 1\leq i \leq n+p$ is an augmented data set, where $\{(y_i^*, x_{i1}^*,\ldots, x_{ip}^*)\}=\{(y_i, x_{i1},\ldots, x_{ip})\}$, $1\leq i\leq n$ and $\{(y_i^*, x_{i1}^*, \cdots x_{ip}^*)\} = \{(0, \lambda_n \mathbf{e}'_{i-n})\}$, $n+1\leq i \leq n+p$, here $\mathbf{e}_i$ is a p-dimensional unit vector with ith element equal to 1. Then $L(\boldsymbol{\beta})$ can be rewritten as

$$\sum_{i=1}^{n}|y_i - \sum_{j=1}^{p}x_{ij}\beta| + \lambda\sum_{j=1}^{p}|\beta_j| = \sum_{i=1}^{n+p}|y_i^* - \sum_{j=1}^{p}x_{ij}^*\beta| \tag{4.2}$$

Thus, $\hat{\boldsymbol{\beta}}(\lambda)$ in LAD-Lasso can be found by using any method for solving linear optimization problem such as the QUANTREG R package if $p \leq$ n. For large p, a very fast and efficient coordinate descent algorithm proposed by [71] can be used. Here the prediction optimal L$_1$ regularization parameter λ can be chosen by minimizing either the

Akaike Information Criterion (AIC score) or the Bayesian Information Criterion (BIC score) defined in the chapter 3.

4.1.2 Asymptotic Properties of the LAD-Lasso

This subsection discuses the conditions, studied by [80], under which the LAD-Lasso has the estimation consistency and also discusses the necessary and sufficient conditions under which the LAD-Lasso estimator is model selection consistent.

4.1.2.1 The estimation consistency when *p* is fixed

Considered a triangular array of observations as in (3.40), Take $\boldsymbol{\beta}^0 = \left(\beta_1^0, \, . \, . \, . \, , \, \beta_p^0\right)'$ as the true model and assume the active set $S_0 = \left\{j \, ; \beta_{n;j}^0 \neq 0, j = 1, \, . \, . \, . \, , p\right\} = \{1, \, . \, . \, . \, , |S_0|\}$ consists of the first $|S_0|$ variables. Let the $\mathbf{X} = \mathbf{X}_n$ the designed matrix with fixed covariates and let the following assumptions.

(A_1) (very general error distributions) The ε_i's are independent and identical distributed has 0 median and continuous positive density *f* in a neighborhood of 0.

(A_2) (identifiability of the regression parameters). With $\boldsymbol{\Sigma}_n \equiv n^{-1}\mathbf{X}_n'\mathbf{X}_n$, there exists a positive definite matrix $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Sigma}_n \rightarrow \boldsymbol{\Sigma}$. If τ_{1n} and τ_{2n} are the minimum and maximum eigenvalues of $\boldsymbol{\Sigma}_n$, there exist constants $0 < \tau_1 < \tau_2 < \infty$ *s.t.* $\tau_1 \leq \tau_{1n} \leq \tau_{2n} \leq \tau_2$ for all *n*

Theorem 4.1. Let $\hat{\boldsymbol{\beta}}_n$ be the LAD-Lasso estimator related to a sequence λ_n in (4.1). If we have the assumptions (A_1) and (A_2), then $\hat{\boldsymbol{\beta}}_n$ has the following asymptotic properties.

a. If $\lambda_n = o(n)$, then $\hat{\boldsymbol{\beta}}_n \rightarrow_P \hat{\boldsymbol{\beta}}^0$, i.e. the LAD-Lasso is consistent estimator.
b. If $(\lambda_n/\sqrt{n}\,) \rightarrow \lambda_0 \geq 0$, then $\hat{\boldsymbol{\beta}}_n$ has the limiting distribution $\sqrt{n}\,(\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}^0) \rightarrow_d$ arg min($\mathbf{V}(\,\mathbf{u})$), where $\mathbf{V}(\,\mathbf{u})$ equal to

$$-2\mathbf{u}'\mathbf{W}^* + f\,(0)\mathbf{u}'\boldsymbol{\Sigma}\mathbf{u} \, + \, \lambda_0\sum_{j=1}^{p}\left[u_j sign\left(\beta_j^0\right) I\left(\beta_j^0 \neq 0\right) + \left|u_j\right| I\left(\beta_j^0 = 0\right)\right]. \quad (4.3)$$

and $\mathbf{W}^*$ is a random vector with a N($\mathbf{0}$, $\boldsymbol{\Sigma}$/4) distribution.

4.1.2.2 The estimation consistency for high dimensions

Let write $p_n = p$ to indicate that p can diverge with n, now with $\mathrm{p} \to \infty$ faster than n i.e. with $\mathrm{p_n/n} \to \infty$ as $\mathrm{n} \to \infty$. Thus, to discuss the identification of model and consistency of the LAD-Lasso estimator, additional assumptions on the sparsity of the model and other regularity conditions are requested.

Let A be any subset of $\{1, \ldots, p\}$ and $\mathbf{X}_A = (\mathbf{x}_j, j \in A)$. For any positive integer $m \leq p_n$, let

$$c_{\min}(m) = \min_{|s_0|=m} \min_{\|v\|_2=1} \frac{1}{n} \mathbf{v}'\mathbf{X}'_A \mathbf{X}_A \mathbf{v} \tag{4.4}$$

and

$$c_{\max}(m) = \max_{s_0=m} \max_{\|v\|_2=1} \frac{1}{n} \mathbf{v}'\mathbf{X}'_A \mathbf{X}_A \mathbf{v}. \tag{4.5}$$

Assume the further assumptions,

(B_1) (Random errors) (A_1) holds.

(B_2) (covariates are bounded) There is a positive constant b_0, *s.t.* $|x_{ij}| < b_0$ $\forall\, i, j$, and $\sum_{i=1}^{n} x_{ij}^2 = n \ \ \forall j.$

(B_3)There is a positive M_1 s.t. $|S_0| \leq M_1 n^2/\lambda_n^2$.

(B_4) There exist constants $0 < c_* < c^* < \infty$ *s.t.*, for any sufficiently large n,

(a) $0 < c_{\min}(\min\{n, p_n\}) \leq c_{\max}(\min\{n, p_n\}) < c^* < \infty$;

(b) $c_{\min}(d_n) > c_*$ for $d_n \leq Mn^2/\lambda_n^2$, where M is a positive constant.

The condition (B_3) is Sparsity of model.

The condition (B_4) is the sparse Riesz condition which controls the range of eigenvalues of covariate matrices of subsets of a fixed number of design vectors $\mathbf{x_j}$. Note that the eigenvalues of the correlation matrix are bounded and that the eigenvalues of any submatrix of the correlation matrix with dimension $O\left(n^2/\lambda_n^2\right)$) are bounded away from zero.

Theorem 4.2 If λ_n^4/n^3 = O(1) and (B_1)−(B_4) hold, then

$$\left\|\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}^0\right\|_2^2 = O\left(\lambda_n^2 \left|S_0\right| n^{-2} c_*^{-2} 2f^{-1}(0)\right) + O_P\left(d_n log\left(2p_n\right) n^{-1} c_*^{-2} 2f^{-1}(0)\right), \qquad (4.6)$$

where $d_n = 2(M_1 + 5c^*/4)(n^2 / \lambda_n^2)$, and $\left|S_0\right|$, M_1, c_*, and c^* are defined in (B_3) and (B_4).

From theorem 4.2, it can be noted that the LAD-Lasso estimator is consistent for suitable values of λ_n even $\log(p_n) = O(n^{\alpha})$ for some $0 < \alpha < 1$.

4.1.3 Model Selection Consistency

4.1.3.1 Model Selection Consistency when *p* is fixed

First, let us to state the definition of the sign consistency of the LAD-Lasso estimator as defined by [80]. The sign consistency is technical and is needed for proving the necessity of the Irrepresentable Condition, which is defined in subsection (3.3.3).

Definition 4.1. A LAD-Lasso estimator is strongly sign consistent if there exists a sequence of λ_n *s.t.* $\lim_{n \to \infty} \mathbf{P}\left(\hat{\beta}_n\left(\lambda_n\right) =_s \boldsymbol{\beta}^0\right) = 1$, where $\hat{\boldsymbol{\beta}}_n\left(\lambda_n\right) =_s \boldsymbol{\beta}^0$ means that $\hat{\boldsymbol{\beta}}_n\left(\lambda_n\right)$ and β^0 have the same sign component-wisely.

Definition 4.2. If $\lim_{n \to \infty} \mathbf{P}\left(\exists \lambda > 0, \hat{\boldsymbol{\beta}}_n\left(\lambda_n\right) =_s \boldsymbol{\beta}^0\right) = 1$, the LAD-Lasso estimator is general sign consistent.

Note sign consistency is stronger than the usual selection consistency, which only requires the zeros to be matched, but not the signs. Now for fixed p, the following theorem provides sufficient conditions under which the LAD-Lasso is strongly sign consistent.

Theorem 4.3. Let p be fixed. Suppose that (A_1), (A_2), and the strong irrepresentable condition are satisfied. Then $\mathbf{P}\left(\hat{\boldsymbol{\beta}}_n\left(\lambda_n\right) =_s \boldsymbol{\beta}^0\right) = 1$ for $\lambda_n = O\left(n^{\pi_2}\right)$, where $(1 + \pi_1)/2 < \pi_2 < 1$ for some $0 < \pi_1 < 1$.

Theorem 4.4. For fixed p, under (A1) and (A2), the LAD-Lasso cannot be general sign consistent if the weak irrepresentable condition does not hold.

From theorems 4.3 and 4.4, it can be noted that the irrepresentable condition is almost sufficient and necessary for sign consistency of the LAD-Lasso.

4.1.3.2 Model Selection Consistency when high dimensions

When $p \gg n$, the assumptions and regularity conditions in (A_2) are inappropriate since $\mathbf{\Sigma}_n$ may not converge as n grows. In this case, some structural conditions on the model are required.

(C_1) (A_1) holds.

(C_2) (B_2) holds.

(C_3) If $b_{n_1} = \min_{j \in S_0} |\beta_j^0|$, then

(a) there exists $0 \le c_1 < 1/2$ *s.t.* $|S_0| = O\left(n^{c_1}\right)$;

(b) there exist positive constants M_0 and $c_2 > c_1$ *s.t.* $n^{(1-c_2)/2} b_{n1} \ge M_0$.

(C4) There exist constants c_* and c^* such that, for any $m_n = O\left(n^{c_1}\right)$, we have

$$0 < c_* < c_{min(m_n)} < c_{max(m_n)} \le c_{max(n)} < c^* < \infty.$$

The condition (C_3) assumes the following:

a) the number of nonzero coefficients increases with n at a slower rate than root n

b) the true nonzero coefficients cannot be too small.

The condition (C_4) assumes that the correlation matrix satisfies the sparse Riesz condition on the rank of $O(|S_0|)$.

The following theorem provides the conditions under which LAD-Lasso is Model Selection

Theorem 4.5. Suppose (C_1)−(C_4) and the strong irrepresentable condition are satisfied. The LAD-Lasso is strong sign consistent even if $p_n = O\left(exp\{n^{c_3}\}\right)$ for $c_3 < \min\{c_2 - c_1, 1 - 2c_1, 1/2\}$. In particular, if $\lambda_n = O\left(n^{(1+c_4)/2}\right)$ with $c_3 < c_4 < \min\{c_2 - c_1, 1 - 2c_1, 1/2\}$, then $\mathbf{P}\left(\hat{\beta}_n(\lambda_n) =_s \beta^0\right) = 1$ as $n \to \infty$. (4.7)

4.2 Robust LARS (RLARS) regression

LARS algorithm can be formulated with only using the correlation matrix as information [81]. Once the correlation matrix is calculated, the actual observations are not required anymore. However, the authors in [81] showed that in the sequencing step, the sequences generated by LARS are not robust against outliers. Therefore, the approaches are based on robust correlations estimates to robustify LARS algorithm.

LARS has been robustified in [81] two different approaches: the plug-in method and the cleaning method. These approaches used robust bivariate correlation estimates which can be computed efficiently using bivariate Winsorization which estimate the correlations and shrink the outliers, respectively. Thus, the influence of potential outliers on computing the sequence of predictors is reduced.

4.2.1 The Plug-In Approach

In the plug-in method, the non-robust estimators mean, variance and correlation in classical LARS are replaced by robust counterparts. The mean and variance are replaced by the median (MED) and the median absolute deviation (MAD) respectively

to robustly standardize the data. Unfortunately, computing robust correlation estimators from the d-dimensional data are computationally inefficient [24]. The robust pairwise approaches were first proposed by [38], where the pairwise correlations are calculated and then assemble to form a correlation matrix. Unfortunately, these approaches are not affine equivariant and therefore are sensitive to two-dimensional outliers. One solution is to use a robust pairwise affine equivariant covariance estimators. A computationally efficient option is a bivariate M-estimator as defined by [82]. Another solution is to compute bivariate correlation estimator from bivariate Windsorized data. These two method will be illustrated in some detail below.

4.2.1.1 M Plug-in

The affine invariant M-estimates of location vector $\boldsymbol{t}$ and scatter matrix $\mathbf{V}^*$ were first proposed by (Maronna in [82]) as robust alternatives to the sample mean vector and covariance matrix. It has been shown that an upper bound for the breakdown point of an M-estimator greater than 1/(p+1) [82] and a general bound of 1/p for a slightly more general class of M-estimators [83].

More preciously, let $\boldsymbol{x}_i,\ i = 1, \dots, n$ be a sample of p-multivariate density f of the form

$$f\left(\boldsymbol{x}_i\right) = \left(det\ \mathbf{V}^*\right)^{-\frac{1}{2}} g\left[(\boldsymbol{x}_i - \mathbf{t})'\mathbf{V}^{*-1}(\boldsymbol{x}_i - \mathbf{t})\right], \tag{4.8}$$

where $g\left(\|\boldsymbol{x}\|_2\right)$ is density in $\Re^p$, then M-estimates of location vector ***t*** and scatter matrix $\mathbf{V}^*$ are defined solution of the system of equations of the form.

$$\frac{1}{n}\sum_i u_1(d_i)(\boldsymbol{x}_i - \boldsymbol{t}) = \mathbf{0}, \tag{4.9}$$

$$\frac{1}{n}\sum_i u_2(d_i^2)(\boldsymbol{x}_i - \boldsymbol{t})(\boldsymbol{x}_i - \boldsymbol{t})' = \mathbf{V}^*, \tag{4.10}$$

where $d_i^2 = (\boldsymbol{x}_i - \mathbf{t})'\, \mathbf{V}^{*-1}(\boldsymbol{x}_i - \mathbf{t}),$ and u_1 and u_2 are functions satisfying a set of following general assumptions:

(A) u_1 and u_2 are nonnegative, nonincreasing, and continuous for $s \geq 0$.

(B) ψ_1 and ψ_2 are bounded. Let $K_i = \text{sub}_{s\geq 0}\psi_i(s)$

(C) ψ_2 is nondecreasing, and strictly increasing in the interval where $\psi_2 < K_2$.

(D) There exist s_0 such that $\psi_2\left(\mathrm{s}_0^2\right) > m$ and that $\mathrm{u}_1(s) > 0$ for $s \leq s_0$ (hence $K_2 > m$).

and include as a particular case the maximum-likelihood estimates, with the functions $\mathrm{u}_1(s) = -s^{-1}\,\mathrm{d}[\log h(s)]/\mathrm{d}s$ and $\mathrm{u}_2\left(s^2\right) = \mathrm{u}_1(s)$, for $s > 0$. Under these assumptions, there exist unique solution for equations (4.9) and (4.10) and the estimates are consistent and asymptotically normal [82].

Authors in [81] used Maronna's bivariate M-estimator of the location vector ***t*** and scatter matrix $\mathbf{V}^*$ that is a highly robust and computationally efficient estimator. The estimators are affine equivariant and have breakdown point 1/3 in two dimensions.

To obtain more simple computations, they used the coordinatewise median as the bivariate location estimate and only solved (4.6) to estimate the scatter matrix and hence the correlation. Then he used the function $\mathrm{u}_2\left(\mathrm{t}^*\right) = \min\left(\mathrm{c}/\mathrm{t}^*, 1\right)$ with $c = 9.21$, the 99 % quantile of a χ_2^2 distribution. After that, the bivariate correlations are assembled to form a $p \times p$ correlation matrix *R*. Finally, this robust correlation matrix is used in LARS.

4.2.1.2 W Plug-in

For very large, high-dimensional data, Huber in [38] introduced the idea of one-dimensional Windsorization of the data to get faster robust correlation estimator. He suggested that classical correlation coefficients be calculated from the transformed data. This approach for the estimation of individual elements of a large-dimension correlation matrix were re-examined by [84].

For n univariate observations $x_1, x_2 \ldots, x_n$, the transformation data can be obtained by

$$u_i = \psi_{\mathrm{HU}}\left((x_i - \mathrm{MED}(x_i))/\mathrm{MAD}(x_i)\right),\ i = 1, 2, \ldots, n, \tag{4.11}$$

where the Huber score function $\psi_{HU}(x)$ is defined as $\psi_{HA}(x) = \min\{\max\{-c, x\}, c\}$and c a tuning constant chosen by the user, e.g., $c = 2$ or $c = 2.5$. This procedure is computationally efficient, but the orientation of the bivariate data makes the outlying observations are only shrunken to the boundary of a $2c \times 2c$ square, as shown in Figure 4.1.

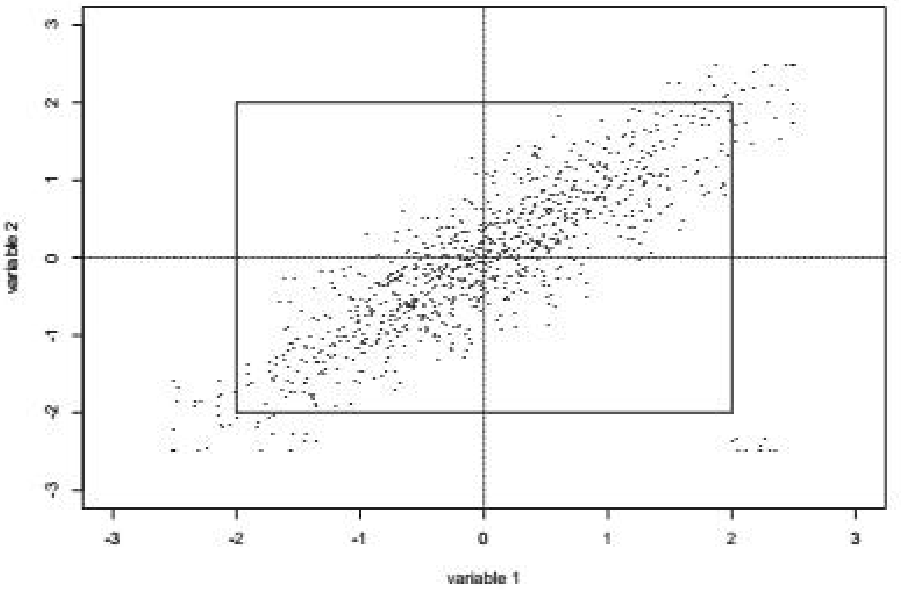

Figure 4.1 Illustration of the limitations of separate univariate Winsorizations (c = 2) when computing robust correlation estimates. The correlation outliers are only shrunken to the boundary of the square.

Authors also in [81] introduced an even faster robust pairwise correlation estimator as a generalization of the univariate Winsorization introduced by [38]. A bivariate Winsorization of robustly standardized bivariate data is based on an initial robust bivariate correlation matrix $\mathbf{R}_0$ and corresponding tolerance ellipse. The outliers are shrunken to the border of this ellipse by using the bivariate transformation.

$$u = \min(\sqrt{c / \mathrm{D}(\boldsymbol{x})},\ 1) \quad \text{with } \boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)'. \tag{4.12}$$

Here $D(\boldsymbol{x}) = \sqrt{\mathbf{x}'\mathbf{R}_0^{-1}\mathbf{x}}$ is the Mahalanobis distance of **x** based on the initial bivariate correlation matrix $\mathbf{R}_0$. The tuning constant c again controls the robustness of the procdure. They proposed to use c = 5.99, the 95% quantile of the χ_2^2 distribution. If the data follows a multivariate normal distribution, the squared Mahalanobis distance follows a χ^2 distribution. Figure 4.2 illustrates bivariate Windsorizations for both the complete data set and the data set excluding the outliers. Bivariate Winsorization shrinks the outliers to the boundary of the ellipse. Correlation outliers are thus appropriately downweighted so that a robust correlation estimate is obtained. The bivariate Winsorized correlation estimate is the classical correlation estimate obtained from the bivariate Winsorized data.

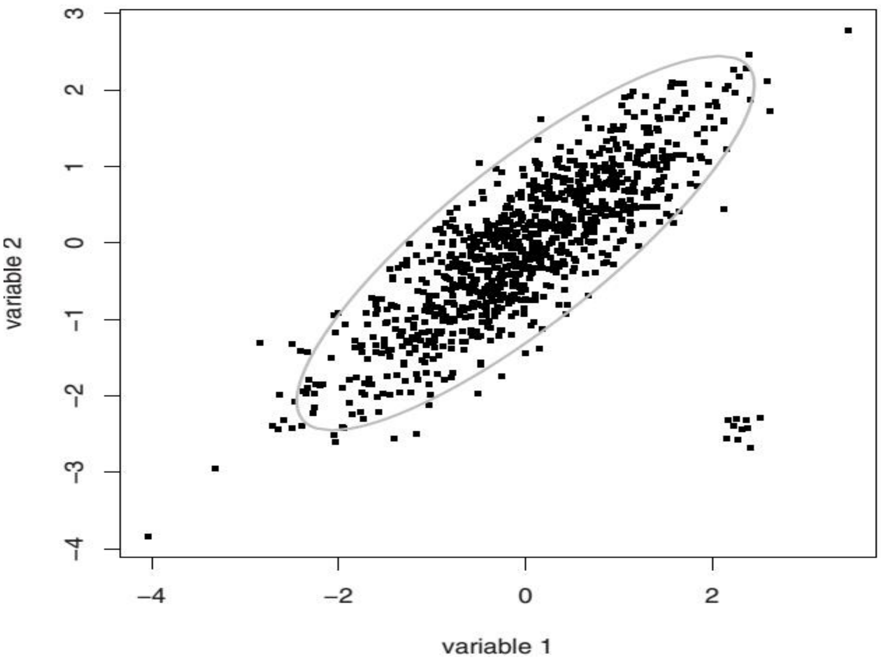

Figure 4.2 Bivariate Winsorization tolerance ellipse, which connects points of equal Mahalanobis distance (2.45) based on the coordinatewise median as robust center and the adjusted Winsorized correlation matrix $\mathbf{R}_0$.

Choosing an appropriate initial correlation matrix $\mathbf{R}_0$ is an important part of the bivariate Winsorization procedure. Hence, Authors also in [81] proposed so-called adjusted Winsorization. This method applies univariate Winsorization to each of the two components of the robustly standardized bivariate data. This method uses two tuning constants to perform univariate Winsorization of the data. That is, for each component, the observations $x_{1j}, \ldots, x_{nj}$, are transformated to $u_{ij} = \min(\max(-c, x_{ij}), c)$; $i = 1, \ldots, n,$ and $j = 1, 2$. However, two different values of the tuning constant c are used. The four quadrants relative to the center zero are considered. The first and larger tuning constant c_1 is used in the two diagonally opposed quadrants that contain the majority of the standardized data. The second tuning constant c_2, which is smaller, is used for the remaining data, in the other two quadrants. They suggested to use $c_1 = 2$ and $c_2 = \sqrt{h}c_1$ where $h = n_2/n_1$, where n_1 is the number of observations in the major quadrants and $n_2 = n - n_1$. The initial matrix $\mathbf{R}_0$ is obtained by computing the classical correlation matrix of the adjusted Winsorized data.

For the same data as in Figure 4.2, Figure 4.3 shows how adjusted Winsorization deals with bivariate outliers, which are now shrunken to the boundary of the squares.

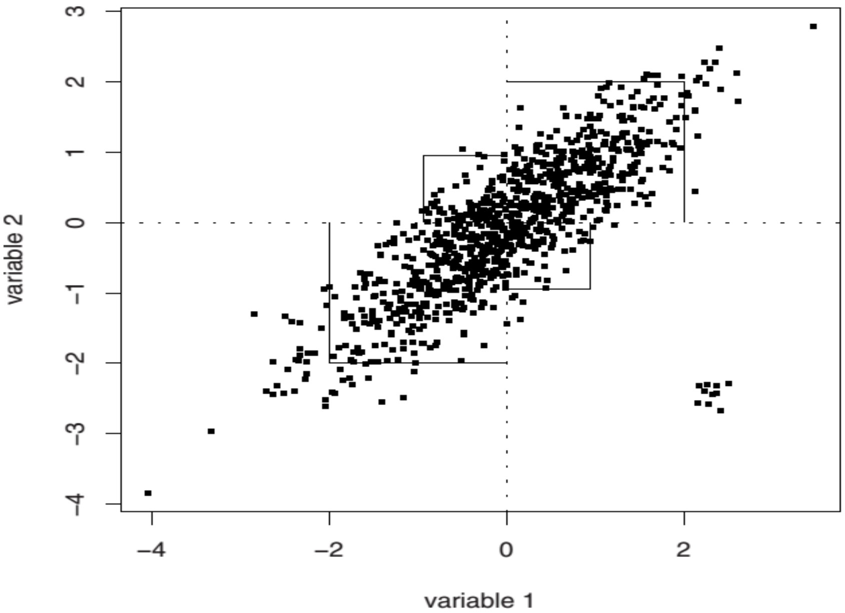

Figure 4.3 Adjusted Winsorization for computing the initial robust correlation estimate R_0 (with $c_1 = 2$ and $c_2 = \sqrt{h}c_1$). The outlying points are shrunken to the edges or corners of the squares.

4.2.2 Data Cleaning Robust LARS

If the dimension p is not too large and the relative sample size is not too small ($p \leq 50$ and $n/p \geq 3$, say), an alternative approach to robustify LARS by shrinking outliers and applying classical LARS to the cleaned data.

Every data point $\boldsymbol{x}_i$ will be replaced by its Winsorized counterpart u_i =min(c/D($\boldsymbol{x}_i$)) in the p dimensional space. The Mahalanobis distance is calculated using a robust initial estimate of the covariance matrix. Then any of the methods explained earlier can be used.

Since the plug-in approach can be used even when the dimension p exceeds the sample size, it is more applicable. The plug-in approach called by [81] RLARS, standing for Robust Least Angle Regression.

4.3 Sparse Least trimmed Squares regression

The least trimmed squares (LTS) estimator has simple definition and highly robust, but it does not has model selection property and cannot be computed for high-dimensional data ($p > n$). To solve these problems, [17] combined the lasso estimator with the LTS estimator and developed the sparse LTS-estimator.

Sparse LTS estimator is a sparse and regularized version of the LTS obtained by adding an L_1 penalty with penalty parameter λ to (3.39),

$$\hat{\boldsymbol{\beta}}_{spLTS} = \arg\min \sum_{i=1}^{h} r_{(i)}^2(\boldsymbol{\beta}) + h\lambda \sum_{j=1}^{p} \left|\beta_j\right|, \tag{4.13}$$

where $r_i^2(\beta) = (y_i - \boldsymbol{x}_i\beta)^2$ denotes the squared residuals and $r^2{}_{(1)}(\boldsymbol{\beta}) \leq . .. \leq r^2{}_{(n)}(\boldsymbol{\beta})$ their order statistics. Here $\lambda \geq 0$ is a penalty parameter and $h \leq n$ the size of the supposed subsample of non-outlying observations.

When $h = n$ the sparse LTS (4.13) yields the Lasso solution (3.39), i.e. sparse LTS can be interpreted as a trimmed version of the Lasso, and can be applied to high-dimensional data.

4.3.1 The breakdown point of the Lasso

We state the following theorem (theorem 1 from [17]) to extract the breakdown point of the sparse LTS estimator.

Theorem 4.6. Let $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ denote the sample and Let $\rho(x)$ be a convex and symmetric loss function with $\rho(0) = 0$ and $\rho(x) > 0$ for $x \neq 0$, and define ***ρ(x)*** := $(\rho(x_1), \ldots, \rho(x_n))'$. With subset size $h \leq n$, consider the regression estimator

$$\hat{\boldsymbol{\beta}} = \arg\min \sum_{i=1}^{h} \boldsymbol{\rho}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + h\lambda \sum_{j=1}^{p} \left|\beta_j\right|, \tag{4.14}$$

where $(\boldsymbol{\rho}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})))_{1:n} \leq \ldots \leq (\boldsymbol{\rho}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}))_{n:n}$ are the order statistics of the regression loss. Then the breakdown point of the estimator $\hat{\boldsymbol{\beta}}$ is given by

$$\varepsilon_n^*(\hat{\boldsymbol{\beta}}\,;\boldsymbol{Z}) = \frac{n-h+1}{n}. \tag{4.15}$$

In particular, applying theorem 4.6 (with subset size $h \leq n$, in which $\rho(x) = x^2$) can extract a finite-sample breakdown point for the sparse LTS estimator $\hat{\boldsymbol{\beta}}_{spLTS}$, and it still equal to $(n - h + 1)/n$.

From formula (4.12), it is noticed that

1. A finite-sample breakdown point for $\hat{\boldsymbol{\beta}}_{spLTS}$ does not depend on the dimension p. A high breakdown point is guaranteed even if p exceeds n.
2. The smaller the value of h, the higher the breakdown point. By taking h small enough, it is even possible to have a breakdown point larger than 50%.

However, this requires to use $h < n/2$, this means that the final estimate model is less statistical efficiency because it is not based on a sufficiently large number of observations. Instead, taking a value of h equal to a fraction α of the sample size, with $\alpha = 0.75$, the final estimate model consider majority of the observations, hence it is good fitting. This option guarantees a sufficiently high statistical efficiency and breakdown point about $1 - \alpha = 25\%$.

4.3.2 The sparse LTS estimator algorithm

The sparse LTS estimator (4.13) can be computed as in the following produce. For a fixed penalty parameter λ, define the objective function

$$Q(H,\boldsymbol{\beta}) = \sum_{x_i \in H} (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + h\lambda\sum_{j=1}^{p}\left|\beta_j\right|, \tag{4.16}$$

which is the L_1 penalized residual sum of squares based on a subsample $H \subseteq \{1, \ldots, n\}$ with $|H| = h$. With

$$\hat{\boldsymbol{\beta}}_H = \underset{\beta}{\operatorname{argmin}}\ Q(H,\boldsymbol{\beta}), \tag{4.17}$$

the sparse LTS estimator is given by $\hat{\boldsymbol{\beta}}_{H_{opt}}$, where

$$\hat{\boldsymbol{\beta}}_{opt} = \underset{H \subseteq \{1,\ldots,n\}:|H|=h}{\operatorname{argmin}}\ Q(H,\boldsymbol{\beta}), \tag{4.18}$$

therefore, the computation of the sparse LTS is equivalent to search for the subset of $h \leq n$ observations whose Lasso fit produces the smallest penalized residual sum of squares. To find this optimal subset, [17] used an analogue of the Fast-LTS algorithm developed by [55], which is faster than the approximating algorithms for the LTS-estimator that do not use the C-step. This algorithm is based on *concentration steps* or C-steps. The C-step at iteration k consists of computing the Lasso solution based on the current subset H_k, with $|H_k| = h$, and constructing the next subset H_{k+1} from the observations corresponding to the h smallest squared residuals. Let H_k denote a certain subsample derived at iteration k and let $\hat{\boldsymbol{\beta}}_{H_k}$ be the coefficients of the corresponding lasso fit. After computing the squared residuals $r_2^k = \left(r_{k,1}^2, \ldots, r_{k,n}^2\right)^2$ with $r_i^2(\boldsymbol{\beta}) = (y_i - \boldsymbol{x}_i \boldsymbol{\beta})^2$, the subsample H_{k+1} for iteration $k+1$ is defined as the set of indices corresponding to the h smallest squared residuals. In mathematical terms, this can be written as

$$H_{k+1} = \left\{ i \in \{1, \ldots, n\} : r_{k,i}^2 \in \{(r_k^2)_{j:n} : j = 1, \ldots, h\} \right\}, \tag{4.19}$$

where $(r_k^2)_{1:n} \leq \ldots \leq (r_k^2)_{n:n}$ are the order statistics of the squared residuals. Let $\hat{\boldsymbol{\beta}}_{H_{k+1}}$ denote coefficients of the lasso fit based on H_{k+1}. Then

$$Q(H_{k+1}, \hat{\beta}_{H_{k+1}}) \leq Q(H_{k+1}, \hat{\beta}_{H_k}) \leq Q(H_k, \hat{\beta}_{H_k}),$$

(4.20)

where the first inequality follows from the definition of $\hat{\beta}_{H_{k+1}}$, and the second inequality from the definition of H_k. From (4.20) using C-step brings the sparse LTS objective function to decrease and if the C-step is applied a sufficient (finite) number of times, a local minimum of the objective function (4.13) is obtained.

The regression model in (2.7) does not include an intercept, so the predictors are all standardized (i.e. all the predictors have mean 0 and variance 1), and the response variable has mean 0 before applying the lasso. However, because the computed means and standard deviations over all observations do not result in a robust method, the authors in [17] applied a different procedure in which each time the sparse LTS algorithm computes a lasso fit on a subsample of size h. First, they computed the means and standard deviations from the own subsample used to center and standardize the variables and the predictors are standardized. The resulting procedure then minimizes (4.10) with squared residuals $r_i^2(\boldsymbol{\beta}) = (y_i - \beta_0 - \mathbf{x}_i' \boldsymbol{\beta})^2$, where β_0 is the intercept. They verified that adding an intercept to the model has no impact on the breakdown point of the sparse LTS estimator of $\boldsymbol{\beta}$.

4.3.3 Reweighted sparse LTS estimator

Assume α denote the proportion of observations from the full sample that we want to maintained in each subsample, i.e., $h = \lfloor (n+1)\alpha \rfloor$. Then $(1-\alpha)$ may be interpreted as an initial guess of the proportion of outliers in the data. This initial guess is typically quite preventative to ensure that outliers do not has impact effect on the results, and this may lead to statistical inefficient result. To increase efficiency, [17] developed a reweighting step that downweights outliers detected by the sparse LTS estimator. When errors are normal distributed, any observation with a standardized residual greater than a certain quantile of the standard normal distribution may be declared as outliers. Since the sparse LTS estimator like the lasso is biased, so the residuals must be centralized. A usual estimate for the center of the residuals is

$$\hat{\boldsymbol{\mu}}_{raw} = \frac{1}{h}\sum_{i \in H_{opt}} r_i , \tag{4.21}$$

where $r_i = (y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}_{spLTS})$, and $\boldsymbol{H}_{\mathrm{opt}}$ is the optimal subset from (4.17). Then the residual scale estimate associated to the raw sparse LTS estimator can be obtained as

$$\hat{\sigma}_{\mathrm{raw}} = k_{\alpha}\sqrt{\frac{1}{h}\sum_{i=1}^{h}(r_c^2)_{i:n}} \ , \tag{4.22}$$

with squared centered residuals $r_c^2 = \left((r_1 - \hat{\mu}_{raw})^2, \ . \ . \ . \ , (r_n - \hat{\mu}_{raw})^2\right)'$, and

$$k_{\alpha} = \left(\frac{1}{\alpha}\int_{-\Phi^{-1}((\alpha+1)+/2)}^{\Phi^{-1}((\alpha+1)+/2)} u^2 d\Phi(u)\right)^{-1/2} , \tag{4.23}$$

Where k_{α} is a factor to ensure that $\hat{\sigma}_{\mathrm{raw}}$ is coincide with the standard deviation at the normal model. This formulation allows to define binary weights

$$w_i = \begin{cases} 1 & \text{if } \left|\frac{r_i - \hat{\mu}_{\mathrm{raw}}}{\hat{\sigma}_{\mathrm{raw}}}\right| \leq \Phi^{-1}(1-\delta) \\ 0 & \text{if } \left|\frac{r_i - \hat{\mu}_{\mathrm{raw}}}{\hat{\sigma}_{\mathrm{raw}}}\right| > \Phi^{-1}(1-\delta) \end{cases} , \qquad i = 1, \ldots, n. \tag{4.24}$$

Then the *reweighted sparse LTS* estimator is given by the weighted Lasso fit

$$\hat{\boldsymbol{\beta}}_{reweighted} = \sum_{x_i \in H} w_i (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \lambda n_w \sum_{j=1}^{p} \left|\beta_j\right|, \tag{4.25}$$

with $n_w = \sum_{i=1}^{n} w_i$ the sum of weights given in (4.24), the reweighted sparse LTS is the lasso fit based on the observations not flagged as outliers. Of course, other weighting schemes may be used. Using the residual center estimate

$$\hat{\mu}_{reweighted} = \frac{1}{n_w} \sum_{i \in H} w_i (y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_{reweighted}), \tag{4.26}$$

We can compute the residual scale estimate of the reweighted sparse LTS estimator as

$$\hat{\sigma}_{reweighted} = k_{\alpha_w} \sqrt{\frac{1}{n_w} \sum_{i=1}^{n} w_i \left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_{reweighted} - \hat{\mu}_{reweighted} \right)^2}, \tag{4.27}$$

where k_{α_w} is the consistency factor from (4.23) with $\alpha_w = n_w/n$.

4.3.4 Choice of the penalty parameter

The suitable value of the penalty parameter λ can be chosen by optimizing the Bayes Information Criterion (BIC), the Akaike Information Criterion (AIC score) or the estimated prediction performance via cross-validation defined in the chapter 3.

4.4 Our method: L$_1$-Penalized MM-Estimator

However, it should be noted that efficiency is an issue with sparse LTS. To obtain a sparse and robust linear model with high efficiency, we propose the L$_1$-Penalized MM-estimation (henceforth MM-Lasso). Our proposed estimator is an approach of MM estimation, which combines high breakdown value estimation with efficient estimation under the normal model.

The L$_1$-Penalized MM-estimation (henceforth MM-Lasso) can be constructed by a three-stage procedure. In the first stage, we compute an initial consistent estimator $\hat{\beta}_0$ with high breakdown point ε_n^* but possibly low normal efficiency. In the second stage, we compute a robust M-scale estimator $\hat{\sigma}$ of the residuals based on the initial estimate. In the third stage, we compute an L$_1$-Penalized M estimator with fixed scale $\hat{\sigma}$; starting the iterations from $\hat{\boldsymbol{\beta}}_0$; and using a loss function that ensures the desired efficiency. Here, efficiency will be loosely defined as similarity with the classical Lasso estimator at the normal model.

Let $\rho_0(r) = \rho_{\mathrm{BI}}(r/k_0)$, $\rho(r) = \rho_{\mathrm{BI}}(r/k_1)$, and assume that each of the ρ-functions is bounded even in the sense of [13]. The scale M estimator (an M-scale for short) $\hat{\sigma}$ satisfies

$$\frac{1}{n-\hat{q}}\sum_{i=1}^{n}\rho_0\left(\frac{r(\boldsymbol{\beta})}{\hat{\sigma}}\right)=b\,. \tag{4.28}$$

Where $\hat{q}$ is the number of non-zero estimated parameters in $\hat{\boldsymbol{\beta}}$ which depends on λ.

To obtain consistency when the errors are normal, the constant b satisfies b = $E_{\Phi}[\rho\ (t^*)]$, with Φ the standard normal distribution. Note that if $\rho(t^*) = {t^*}^2$ and b = 1 then $\hat{\sigma}$ is the residual standard deviation . The MM-Lasso is defined with

$$L(\mathbf{x}, y, \boldsymbol{\beta}) = \hat{\sigma}^2\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right)+n\lambda\sum_{j=1}^{p}\left|\beta_j\right| \tag{4.29}$$

where the factor $\hat{\sigma}^2$ before the summation is employed to make the estimator coincide with the classical one when $\rho(t^*) = {t^*}^2$. Let ρ satisfy $\rho \le \rho_0$, from [49] it is easy to show if $\hat{\boldsymbol{\beta}}$ satisfies L(x, y, $\hat{\boldsymbol{\beta}}$) ≤ L(x, y, $\hat{\boldsymbol{\beta}}_0$), then $\hat{\boldsymbol{\beta}}$'s ε_n^* is not less than that of $\hat{\boldsymbol{\beta}}_0$. The value of k_0 should be chosen in order to attain high breakdown point of the MM-Lasso. The choice of k_1 will to determine asymptotic efficiency of the estimate without affecting its breakdown point. In order to let $\rho \le \rho_0$, we must have $k_1 \ge k_0$; the larger the k_1 is, the higher efficiency the MM-Lasso can attain at the normal distribution.

4.4.1 IRLS Algorithm

The penalty function in equation (3.39) is convex but it is a non-differentiable function, hence it is difficult to obtain analytic form solution of equation (3.39). Here we can obtain an approximate closed form solution as in [2]:

$$\hat{\boldsymbol{\beta}}_{lasso} = \arg\min\sum_{i=1}^{n}\left(y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right)^2 + n\lambda\sum_{j=1}^{p}\left(\beta_j^2\big/\left|\beta_j\right|\right) \tag{4.30}$$

Iteratively re-weighted least squares (IRLS) algorithm for the Lasso estimate in equation (4.30) can be obtained by computing the ridge regression iteratively as:

$$\hat{\boldsymbol{\beta}}_{lasso}^{(i+1)} = (\mathbf{X}'\mathbf{X} + \lambda\boldsymbol{\Lambda}^{(i)})^{-1}(\mathbf{X}'\boldsymbol{y}), \tag{4.31}$$

where $\mathbf{\Lambda}^{(i)}$ is the generalized inverse (pseudo-inverse) of matrix = diag $\left\{\left|\boldsymbol{\beta}_{lasso,1}^{(i)}\right|,\left|\boldsymbol{\beta}_{lasso,2}^{(i)}\right|,\ldots,\left|\boldsymbol{\beta}_{lasso,p}^{(i)}\right|\right\}$ and i = 0,1... is the iteration number. Similarly, an iteratively re-weighted least squares (IRLS) algorithm for the MM-Lasso estimate. Define

$$\psi(t^*) = \rho'(t^*),\ W\,(t^*) = \frac{\psi(t^*)}{t^*}, \tag{4.32}$$

Let

$$t_i^* = \frac{r_i}{\hat{\sigma}},\ w_i = \frac{W\,(t^*)}{2}, \tag{4.33}$$

$$\boldsymbol{w} = (w_1, w_2, \ldots, w_n)',\ \boldsymbol{W} = diag(\boldsymbol{w}). \tag{4.34}$$

by differentiating of (4.29) with respect to $\boldsymbol{\beta}$ and setting the derivative to zero, one gets,

$$\boldsymbol{w}' = (y - X\,\hat{\boldsymbol{\beta}}) = 0 \tag{4.35}$$

and

$$\hat{\boldsymbol{\beta}}_{MM-lasso}^{(i+1)} = (\mathbf{X}'\boldsymbol{W}^{(i)}\mathbf{X} + \lambda\mathbf{\Lambda}^{(i)})^{-1}(\mathbf{X}'\boldsymbol{W}^{(i)}\boldsymbol{y}). \tag{4.36}$$

Since for the chosen ρ, $W(t^*)$ is a decreasing function of $|t^*|$, observations with larger residuals will receive lower weight w. The iteration will stop until a maximum number is reached or the difference between two successive iteration steps is small enough.

The following is the procedure to obtain the estimator $\hat{\boldsymbol{\beta}}$

1) A high breakdown estimator is used to find an initial estimate $\hat{\boldsymbol{\beta}}_0$ (we choose sparse LTS estimator in [17]). Using this estimate the residuals, $r_i\,(\hat{\boldsymbol{\beta}}_0) = y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_0$, are computed, for $1 \le i \le n$.

2) Using these residuals from the robust fit, an M-estimate of scale $\hat{\sigma}$ with high break down ε_n^* is computed from (4.28).

3) At each iteration with $\hat{\sigma}$ remains fixed throughout, calculate residuals $r_i^{(j-1)}$ and associated weigh $\boldsymbol{w}\left(r_i^{(j-1)}\right)$ according to the weight function.

4) Solve the following for iteratively re-weighted least squares (IRLS) equation,

$$\hat{\beta}^{(j)}_{MM-lasso} = (\mathbf{X}'\boldsymbol{W}^{(j-1)}\mathbf{X} + \lambda\mathbf{\Lambda}^{(j)})^{-1}(X'\boldsymbol{W}^{(j-1)}\boldsymbol{y}), \tag{4.37}$$

Steps 3) and 4) are repeated until $\frac{\left|r_i^{(j)} - r_i^{(j-1)}\right|}{r_i^{(j-1)}}$ becomes less than tolerance.

4.4.2 Weight Functions and Choosing the Constants

Several types of weight functions are proposed for IRLS algorithm in literature. Each set of functions given includes tuning constants, which allow for the shape of the function to be slightly altered. Beaton and Tukey [47] proposed the IRLS algorithm with Tukey's bisquare function that enables to remove the influence of extreme outliers completely from the estimation.

$$\rho_{BI}(t^*) = \begin{cases} \frac{k}{6}\left[1-(1-(t^*/k)^2)^3\right] & if \ |t^*| \le k, \\ \frac{k}{6} & if \ |t^*| > k. \end{cases} \tag{4.38}$$

$$\psi_{BI}(t^*) = \begin{cases} (1-(t^*/k)^2)^2 & if \ |t^*| \le k, \\ 0 & if \ |t^*| > k. \end{cases} \tag{4.39}$$

where $\psi_{BI}(t^*) = \rho'_{BI}(t^*)$ is bisquare score function , and the value k for bisquare function is a *tuning constant.* In particular, the value k_0 =2.937 such that the asymptotically consistent scale estimate $\hat{\sigma}$ has the breakdown value of 25%, while the value k_1 = 3.44 yields 0.85 asymptotic efficiency at the normal model when $\lambda = 0$ [49].

However, the scale estimate $\hat{\sigma}$ requires a correction for high dimensional data. According to [85], there are two problems appear when fitting a standard MM estimator to data with a high ratio p/n:

(1) The scale based on the residuals from the initial regression estimator underestimates the true error scale.

(2) Even if the scale is correctly estimated, the actual efficiency of the MM estimator can be much lower than the nominal one. For this reason, $\hat{\sigma}$ is corrected using (formula (9) in[85]) as

$$\tilde{\sigma} = \frac{\hat{\sigma}}{1-(k_1+k_2/n)\hat{q}/n} \text{ with } k_1 = 1.29,\ k_2 = -6.02. \tag{4.40}$$

4.4.3 Choosing the Penalty Parameter

We propose to select λ by the estimated prediction error of MM-Lasso for different values of λ via cross-validation. We can use the k-fold cross validation process, which requires recomputing the estimate k times. For k = n ("leave-one-out" LOOCV) we can use an approximation to avoid recomputing.

Call $\hat{y}_{-i}$ the fit of y_i computed without using the i-th observation; i.e., $y_{-i} = x_i' \hat{\beta}^{(-i)}$, where $\hat{\beta}^{(-i)}$ is the MM-Lasso estimate computed without observation *i*. Then a first-order Taylor approximation of the estimator yields the approximate prediction errors

$$r_{-i} = y_i - \hat{y}_{-i} \approx \left(1 + \frac{W(t_i^*)h_i}{1 - h_i \psi'(t_i^*)} \right) \tag{4.41}$$

with

$$h_i = \boldsymbol{x}_i' \left(\sum_{i=1}^{n} \psi'(t^*) \boldsymbol{x}_i \boldsymbol{x}_i' + 2\lambda \boldsymbol{\Lambda}^{(i)} \right)^{-1} \boldsymbol{x}_i \tag{4.42}$$

where ψ and t_i are defined in equations (4.32), (4.33) , $\boldsymbol{x}_i$ is the i-th row of $\mathbf{X}$ and $\boldsymbol{\Lambda}^{(i)}$ is the generalized inverse (pseudo-inverse) defined in equation (4.31) . Given the prediction errors $\boldsymbol{r}_{-} = (r_{-1}, \dots, r_{-n})'$, we compute a robust mean squared error (MSE) as $\tau(\boldsymbol{r}_{-})^2$, where τ is a "τ-scale" with tuning constant $c_\tau = 5$ [86], and choose the λ minimizing this MSE.

CHAPTER 5

SIMULATION

5.1 Background of simulation

5.1.1 Systems

A system is a set of elements (components, entities, factors, members, etc.), and each element is characterized by a set of attributes which have logical or numerical values. These elements are related by some form of interaction which act together to achieve some objective or purpose. There are two relationships; internal and external relationships. The internal relationships connect the elements within the system, while the external relationships connect the elements with the environment, that is, with the world outside the system [15, 16].

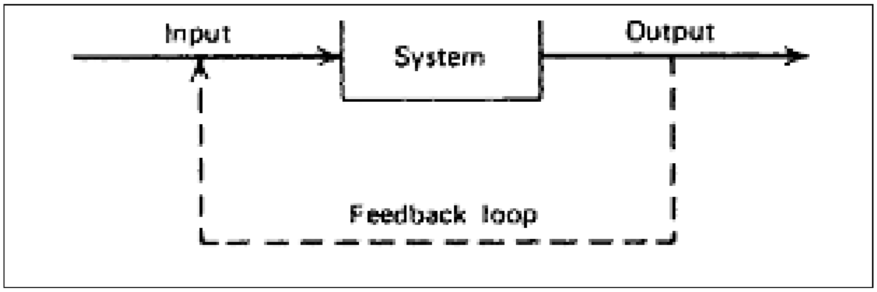

Figure 5.1 Graphical representation of a system

The system is influenced by the environment through inputs, outputs and feedback mechanisms. The system contains feedback mechanisms if can react to change its own state. A nonfeedback system lacks this characteristic. The system has the ability to maintain internal steady-state despite a changing external environment. The attributes of the system elements define its state. When the behavior of the elements cannot be predicted exactly, it is useful to take random observations from the probability distributions and to average the performance of the objective. A system is in equilibrium or in the steady state if the probability of being in some state does not vary in time.

Systems can be classified in a different of ways. There are natural and artificial systems, adaptive and nonadaptive systems closed and open systems.
An adaptive system vs. nonadaptive system; an adaptive system reacts to changes in its environment, whereas a nonadaptive system does not.
Natural vs. artificial systems; a natural system exists as a result of processes occurring in the natural world and an artificial system owes its origin to human activity [87, 88].
Static vs. dynamic, static system; has structure but no associated activity and a dynamic system involves time-varying behavior for complex systems. It deals with internal feedback loops and time delays that affect the behavior of the entire system [89, 90].
Open-loop vs. closed-loop systems; in all systems there will be an input and an output. Inputs are variables that influence the behavior of the system and outputs are variables, which determined by the system and may influence the surrounding environment. An open-loop system cannot control or adjust its own performance but a closed-loop system controls and adjusts its own performance in response to outputs by the system through feedback.

5.1.2 Models

Modeling is the process of building a model; a model can be defined as representation of the construction and working of some real system. The importance of models and model-building has been discussed by [91]. One purpose of a model is to enable the analyst to predict the effect of changes to the system. An important issue in modeling is model validity. Model validation techniques include simulating the model under known input conditions and comparing model output with system output. Thus, while building a model it must be taken to ensure that it remains a valid representation of the problem
To achieve this purpose, a model necessarily embodies elements of two conflicting attributes-realism and simplicity [92]. In other words, the model should provide as a reasonably close approximation to the real system and integrate most of the important aspects of the system. On the other hand, the model must not be so complex that it is impossible to understand and manipulate. A good model can make tradeoff between

realism and simplicity. Simulation practitioners recommend building a model iteratively.

Indeed, [93] provided a definition of a model as “a representation of the system under study, a representation which lends itself to use in predicting the effect on the system’s effeteness of possible changes in the system. (p. 155)”. Based on this definition [93, 94] described many types of models:

1. Iconic models: Those that pictorially or visually represent certain aspects of a system. In iconic models, the relevant properties of real thing are represented by the properties themselves, usually with a change of scale.
2. Analog models: Those that employ one set of properties to represent some other set of properties. They are more abstract than iconic models but can are easier to manipulate and can represent dynamic situations.
3. Symbolic models: Those that require mathematical forms or symbols to represent the variables and their interrelationships between them. Symbolic models or abstract models, can used letters, numbers and other types of symbols.

Generally, the mathematical model is used for a simulation study, which is developed with the help of simulation software. Mathematical models can be classified in many ways, such as static (that do not explicitly consider time-variation) or dynamic (time-varying interactions among variables are taken into account).

Another classification of mathematical model is deterministic versus stochastic models. In a deterministic model, all mathematical and logical relationships between the elements are fixed. Consequently, these relationships completely determine the solutions. In a stochastic model, at least one variable is probabilistic. Typically, simulation models are stochastic and dynamic.

The mathematical model of the system is constructed by combining the formulae describing the behavior of the various components in the systems (elements) with the formulae describing how they interact (the interconnections). The next phase after constructing a mathematical model for the problem under consideration is to develop computer-based procedure that derives a solution from this model. Mathematical models can be solved analytically or numerically. An analytic solution is usually obtained directly from its mathematical representation in the form of formula; hence, analytic solution is mathematical expression that yields a value after suitable substitution of parameter values. Whereas, a numerical solution is generally an approximate solution because it based on the substitution of numerical values for the variables and parameters of the model. The numerical methods may be iterative, that is, each successive step in the solution uses the results from the previous step [95]. The Newton-Raphson method, or Newton Method, is a powerful iterative technique for solving equations numerically. It is based on the simple idea of linear approximation. It can be used for approximating the root of a nonlinear equation. Author of [96] considers two special types of numerical methods; simulation and the Monte Carlo methods.

5.1.3 Simulation, and the Monte Carlo Methods

Simulation was defined in ([97], p.3) as "Simulation is a numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behavior of a business or economic system (or some component thereof) over extended periods of real time."

Simulation is, in its wide sense, a technique for performing sampling experiments on the model of the system [98]. Considering this general definition of simulation, simulation can be successfully used in many situations. The following four reasons for the application of simulation derived by [96, 97]

1) First, it may be either impossible or extremely expensive to obtain data from certain processes in the real world.
2) The observed system may be so complex that it cannot be described in terms of a set of mathematical equations for which analytic solutions are obtainable.
3) Even though a mathematical model can be formulated to describe some system of interest, it may not be possible to obtain a solution to the model by straightforward analytic techniques.
4) It may be either impossible or very costly to perform validating experiments on the mathematical models describing the system.

Simulation is indeed an invaluable and very multilateral tool in those problems where analytic techniques are inappropriate. However, simulation drawbacks and limitations as the following

1. Simulation is an imprecise technique. It does not provide exact results but only statistical estimates and it only compares alternatives rather than generating the optimal one.

2. Simulation is also a slow and costly way to study a problem. It usually requires a large amount of time and great expense for analysis and programming.
3. Finally, simulation yields only numerical data about the performance of the system, and sensitivity analysis of the model parameters is very expensive. The only possibility is to conduct series of simulation runs with different parameter values.

However, the simulation in a narrow sense, or stochastic simulation, is defined by [97] as experimenting with (abstract) model over time; this experimenting includes sampling stochastic variates from probability distribution. Therefore, stochastic simulation is actually a statistical sampling experiment with the model. [97] refers to this kind of simulation as stochastic simulation or Monte Carlo Simulation because random numbers are used. Random numbers are essentially independent random variables uniformly distributed over the unit interval (0, 1).

5.2 The R Package simFrame

In model-based simulation, datasets are generated repeatedly from a distributional model or a mixture of distributions. In each successive step, certain quantities of interest are computed for evaluation of statistical parametric methods that make theoretical assumptions about the underlying data comparisons. For example, outlier detection methods either assume a known underlying distribution of the observations (typically a multivariate normal distribution is assumed) [24, 99, 100] or, at least, they are based on statistical estimates of unknown distribution parameters [101].

SimFrame proposed by [102] is a general statistical simulation R package, but particularly, it is developed for simulation studies in survey statistics. It is focused on simulations involving typical data problems such as outliers and missing values. Therefore, certain proportions of the data may be contaminated or set as missing in order to investigate the quality and behavior of, e.g., robust estimators or imputation methods which are mainly used in the social sciences and discusses [103]. In addition, an appropriate plot method for the simulation results is selected automatically depending on their structure.

5.2.1 Object-oriented programming and S4

The R package simFrame is implemented and developed based on object-oriented programming (OOP) language, which is directly influences the way in which we view the world. Thus, complex problems can be solved in the same manner as they are solved in real-world situations.

5.2.1.1 Classes and objects

In OOP, a collection of interacting objects and classes are used for representing the concepts of abstraction and encapsulation rather than a set of functions. So, the development of programs using object model is known as object-oriented development.

The mapping of abstraction to a program is shown in Figure. 5.2.

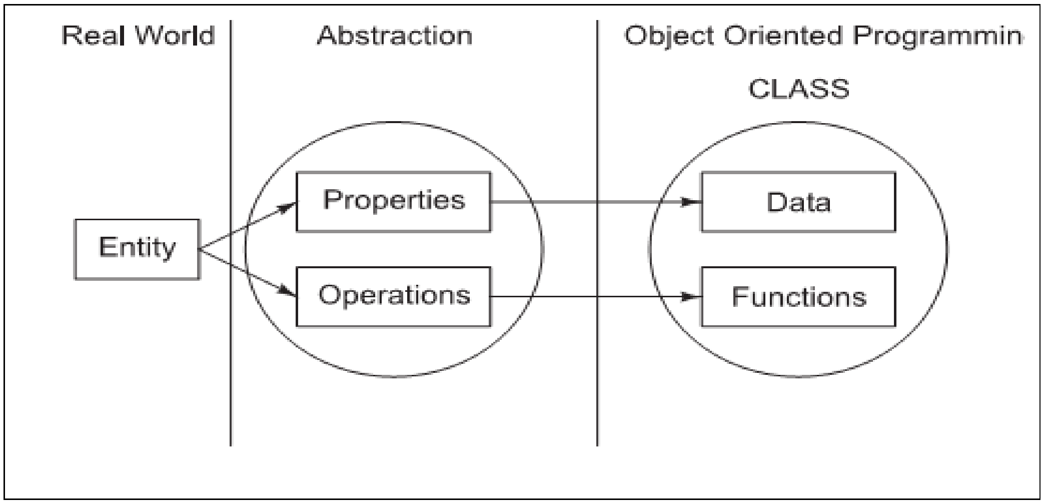

Figure 5.2 Mapping real world entity to object oriented programming

The software structure that supports data abstraction is known as class. A class is a data type capturing the essence of an abstraction and hence it cannot be directly manipulated. The class is a prototype or a model that defines different features. A feature may be a data or an operation. Data are represented by instance variables or data variables in a class. The operations are also known as behaviors, methods, or functions.

For example, car represents a class (a model of vehicle) and there are a number of instances of car. Each instance of car is an object and the class car does not physically mean a car. Actually, each object in an object-oriented system corresponds to a real-world thing, which may be a person, or a product, or an entity. In other words, the properties of these objects are defined by classes and their behavior and interactions are modeled with generic functions and method.

5.2.1.2 Concepts of object-oriented programming

The more important concepts of object-oriented programming are as follows:

- Encapsulation
- Data Abstraction
- Inheritance
- Polymorphism

Encapsulation: Encapsulation is mechanism, by which you combine code and the data it manipulates into a single unit, Encapsulation provides a layer of security around manipulated data, protecting it from external interference and misuse.

Data Abstraction: Real-world objects are very complex and it is very difficult to capture the complete details. Hence, OOP uses the concepts of abstraction and encapsulation to identify essential information system requirements while simultaneously postponing or eliminating non-essential aspects. An abstraction intentionally ignores some qualities, attributes, or functions of an information system in order to focus on the essential attributes and behavior.

Inheritance: Class inheritance is the most important concept in object-oriented programming. It involves the creation of new classes, also called superclasses from existing classes superclasses. The new classes can inherit the properties and behavior from their superclasses. Thus, it allows the extension and reuse of existing code, without having to repeat or rewrite the code from scratch, which is the main advantage of inheritance. In addition, subclasses may have additional properties and behavior, so in this sense they extend their superclasses.

Polymorphism: Polymorphism allows an object to be processed differently by data types and/or data classes. It means that the same operation which is a procedure or transformation that an object performs, may behave differently for different classes. An operation is a procedure or transformation that an object performs.

5.2.1.3 S4 Classes and Methods

The S language, principally, has been designed by [104] who received the ACM System Software award for S, the only statistical software to receive this award. Now S has two implementation: S-plus is commercial, R is open-source system based on the S language. R's international support and the thousands of packages and other contributions have made it.

S4 is the 4th version of S [104]. The main characteristic of S4 compared to S3 is the development of functions which allow to consider S as an object language[1]. By extension, S4 stand for object oriented programming with S.

In S4, properties of objects are stored in slots and can be accessed or modified with the a special operator, @, or the slot() function. Here, Base or virtual class is an abstract class (because it has a pure virtual function), so no objects of class can be created. It is used only to share code. Furthermore, class unions are special virtual classes with no slots. Conceptually, a generic function extends the idea of a function in R by allowing different methods to be selected corresponding to the classes of the objects supplied as arguments in a call to the function. The generic function should specify:

- The overall purpose and meaning; that is the function being performed, in the informal sense of that word.
- The formal arguments, along with a signature that specifies which arguments can be used to select the actual method to be used.

These methods are stored within the generic function according their signature, which assign classes to the formal arguments.

5.2.2 Design of the SimFrame

The fundamental design principle of simFrame is that the behavior of functions can be determined in terms of control objects. A collection of such control objects, including a function to be applied in each iteration, is simply plugged into a generic function called runSimulation() , which then performs the simulation experiment. This allows to easily switch from one simulation design to another by just plugging in different control

[1] allow to consider and not transforms into. In any case, R is not an object oriented language, it remains a traditional language interpreted with a possible further cover.

objects. The user does not have to program any loops for iterations or collect the results in a suitable data structures.

The Unified Modeling Language (UML) [105] is a standardized modeling language used in software engineering. It provides a set of graphical tools to model object-oriented programs. A class diagram visualizes the structure of a software system by showing classes, attributes, and relationships between the classes. Figure 5.3 shows a slightly simplified UML class diagram of simFrame.

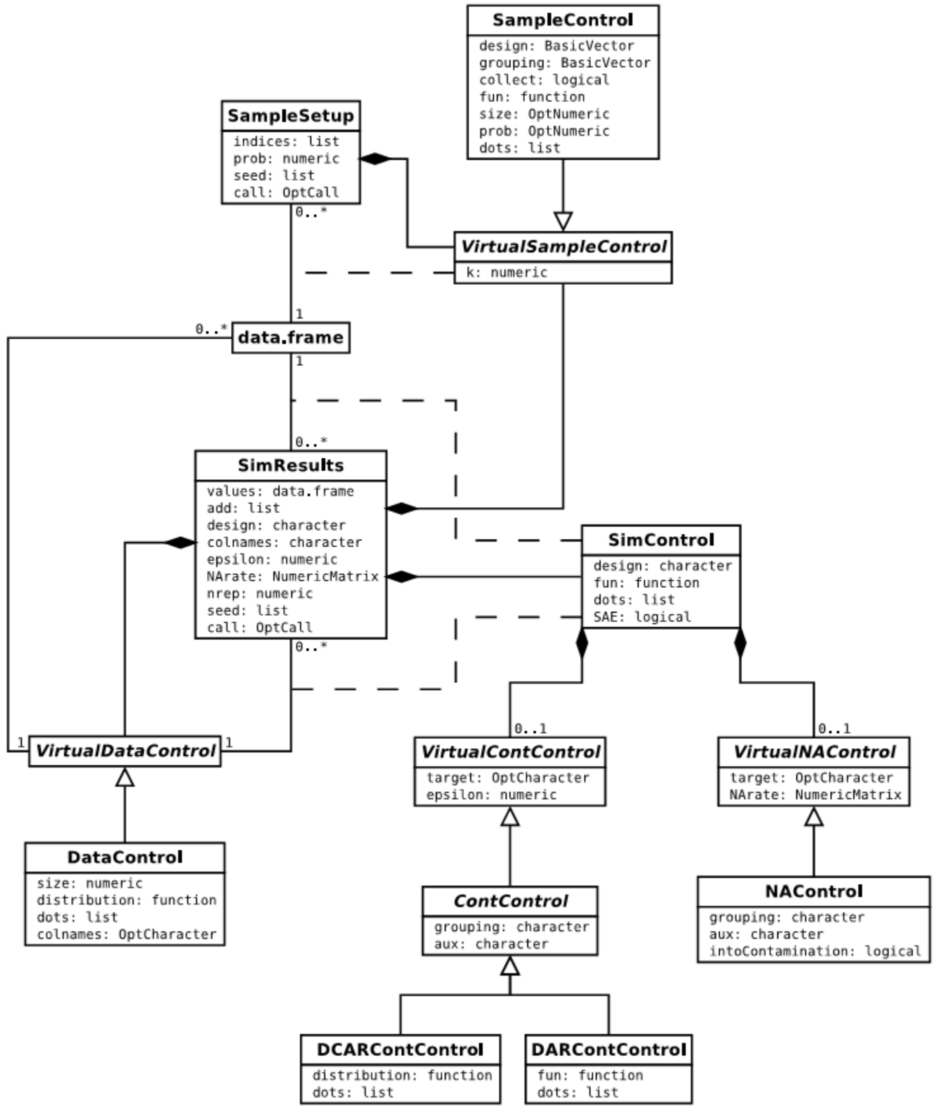

Figure 5.1 Slightly simplified UML class diagram of **simFrame** [102].

5.2.3 Implementation

R is a free software environment for statistical computing. It is an environment incorporating an implementation of the S programming language, which is powerful, flexible and has excellent graphical facilities. R is rotten at iterative algorithms that require loops iterated many times, this makes R has low computing. Fortunately, R includes a well-developed programming language and provides interfaces to many others, including the fast low-level languages C and Fortran. This a way to get all the speed advantages of C or Fortran with most of the convenience of R is to write the inner loop in C and call it from R. The S4 system complies with all requirements for an object-oriented framework for statistical simulation. Thus, most of simFrame is implemented as S4 classes and methods, except some utility functions and some C++ code [102].

Method selection for generic functions is based on control classes, which in most cases provides the interfaces for extensions by developers. The user can not call most of these generic functions directly. The idea of the framework is to define a number of control objects and to supply them to the function runSimulation(). This function performs the complete simulation experiment and calls the other functions internally. Moreover, like R, simFrame has the data handling and manipulation techniques and data are typically stored in a data.frame.

5.2.3.1 Model-based data

Statistical methods usually assume that the data majority originates from multivariate distribution, e.g., outlier detection methods in multivariate statistics usually assume that the majority of the data follow a multivariate normal distribution. Consequently, such methods are typically tested in simulations on data coming from a certain theoretical distribution. The generation of data from a distributional model is handled by control classes inheriting from the class union (which is a special virtual class with no slots) VirtualDataControl. The user can extend the framework by extending this virtual class. A simple control class already implemented in simFrame is DataControl. It consists of the following slots (see also Figure 5.4).

size**:** The number of observations to be generated.

distribution**:** A function for generating the data, e.g., rmvnorm in package ***mvtnorm*** [106, 107] for data following a multivariate normal distribution.

dots**:** Additional arguments to be passed to distribution.

In a model-based simulation study, such a control object is then used by the framework in repeated internal calls of the generic function generate(control, ...) .

The following example demonstrates how to define a control object for generating data from a multivariate normal distribution.

```
R> library("mvtnorm")

R> dc <- DataControl(size = 10, distribution = rmvnorm, dots =

        + list(mean = rep(0, 2), sigma = matrix(c(1, 0.5, 0.5, 1), 2, 2)))
```

Although the function generate( ) is designed to be called internally by the simulation framework, we can use it as a general covering function for data generation in other contexts.

5.2.3.2 Contamination

In addition, the package has been designed with special emphasis on simulations involving typical data problems such as outliers and missing values. It offers mechanisms to contaminate the data and insert missing values so that the influence of these data problems on statistical methods can be investigated, or that outlier detection or imputation methods can be evaluated.

In simulation studies, to study the influence of the outliers on the robust estimators and evaluate robust statistical methods, a certain part of the data needs to be contaminated.

In robust statistics, the distribution F of contaminated data is typically modeled as a mixture of distributions.

According to [108, 109] outliers may be modeled by a two-step process in simulation studies

1. Select the observations to be contaminated. The probabilities of selection may or may not depend on any other information in the data set.
2. Model the distribution of the outliers. The distribution may or may not depend on the original values of the selected observations.

In simFrame, contamination is implemented based on control classes inheriting from VirtualContControl. Figure 5.3 displays the full hierarchy of the available control classes for contamination.

The distribution of the contaminated data in simulation experiments may or may not depend on the original values. Similar to model-based data generation, the control class DCARContControl supports specifying a distribution function for generating the contamination. DCAR stands for distributed completely at random; in this case, the contamination process is independent of the original data. If we select a variable for grouping then the same values are used for all observations in the same group. DCARContControl can extend ContControl by the following slots:

distribution**:** A function for generating the data for the contamination, e.g., rmvnorm in package ***mvtnorm*** for a multivariate normal distribution.

dots**:** Additional arguments to be passed to distribution.

Conversely, the control class DARContControl realizes contamination based on the original values. DAR thereby stands for distributed at random. An arbitrary function may be used to modify the data. To do so, the original values of the observations to be contaminated are passed as its first argument. Thus, the following slots are available in addition to those from ContControl:

fun**:** A function generating the values of the contaminated data based on the original values.

dots**:** Additional arguments to be passed to fun.

5.2.4 Extending the framework

Indeed, the S4 implementation of the R package **simFrame** can provide clear interfaces for user-defined extensions. In order to extend the framework, developers can implement custom control classes and the corresponding methods.

5.2.4.1 Model-based data

SimFrame has general control class DataControl. For user defined data generation models, it often suffices to implement a function and use it as the distribution slot in the DataControl object. This function should have the number of observations to be generated as its first argument, as illustrated in the code skeleton in Figure 5.4 (*top*). In this manner, the name of the argument is not important. Furthermore, the function should return an object that can be coerced to a data.frame. But if more specialized data generation models are required, the framework can be extended by defining a control class extending VirtualDataControl and the corresponding method for the generic function generate(). If the user wants to use a specific distribution or mixture of distributions, he may use a different control class. Figure 6 (*bottom*) presents the code skeleton for such an extension.

```
myDataGeneration <-   function (size , ...) {
# computations
}
```

```
setClass (" MyDataControl ",
        # class definition
       contains = " VirtualDataControl ")
setMethod (" generate ",
       signature ( control = " MyDataControl "),
       function ( control ) {
              # method definition
       })
```

Figure 5.4 Top: Code skeleton for a user-defined data generation method. Bottom: Code skeleton for extending model-based data generation with a custom control class and the corresponding method for generate().

5.2.4.2 Contamination

The control classes DCARContControl and DARContControl can cover many contamination models. But we can add other contamination models by defining a control class inheriting from VirtualContControl and the corresponding method for contaminate() as shown in Figure 5.5. It can be noted that VirtualContControl has the slots target and epsilon for selecting the target variable(s) and contamination level(s), respectively.

```
setClass (" MyContControl ",

         # definition of additional properties
```

```
        contains = " VirtualContControl ")
setMethod (" contaminate ",
        signature (x = " data . frame ", control = " MyContControl "),
        function (x, control , i) {
         # method definition
        })
```

Figure 5.5 Code skeleton for a user-defined control class for contamination and the corresponding method for contaminate( ).

5.3 Simulation Study

To investigate the behavior of our robust estimator MM-Lasso, a simulation study for comparing the performance of various sparse estimators are performed in R (R Development Core Team, 2011) with package simFrame[102], which is a general framework for simulation studies in statistics.

As in [17] we make a comparison with the lasso, the LAD-Lasso, robust least angle regression (RLARS) and Sparse LTS with reweighted step. Sparse LTS is evaluated for the subset size $h = \lfloor (n+1)0.75 \rfloor$ to guarantee a breakdown point of 25%. All computations are carried out in R version 3.1.2 (R Development Core Team, 2011) using the packages robustHD [110] for sparse LTS and RLARS, quantreg [111] for the LAD-Lasso and lars [112] for the Lasso. We implemented MM-Lasso by using C programming language.

For every generated sample, an optimal value of the shrinkage parameter λ is selected. The penalty parameters for MM-Lasso are chosen using k-fold cross validation process as described in subsection (4.4.3), and the other methods are optimized via BIC as described in [17].

5.3.1 Sampling Schemes

In this study, we take the three configurations from [17] to represent low, moderate and high dimensional data. Firstly in the case of $n > p,$ we create a linear model. From $k = 6$ latent independent standard normal variables, L_1 , L_2 , ... , L_k and an independent normal error variable e with standard deviation σ, the response variable y is constructed as

$$y = L_1 + L_2 + \cdot \cdot \cdot + L_k + \sigma\boldsymbol{\varepsilon}, \qquad (5.1)$$

The value of σ is chosen so that the signal to noise ratio is equal to 3. A set of $p = 50$ candidate predictors is then constructed as follows. Let $e_1, ..., e_p$ be independent standard normal variables and let

$\boldsymbol{x}_i = L_i + \tau e_i, i = 1, ..., k$

$\boldsymbol{x}_{k+1} = L_1 + \delta e_{k+1}$

$\boldsymbol{x}_{k+2} = L_1 + \delta e_{k+2}$

$\boldsymbol{x}_{k+3} = L_2 + \delta e_{k+3}$

$\boldsymbol{x}_{k+4} = L_2 + \delta e_{k+4}$

$\vdots$

$\boldsymbol{x}_{3k-1} = L_k + \delta e_{3k-1}$

$\boldsymbol{x}_{3k} = L_k + \delta e_{3k}$

and $\boldsymbol{x}_i = e_i$, $i = 3k + 1, ..., p$

The constants $\delta = 5$ and $\tau = 0.3$ are chosen so that corr($\boldsymbol{x}_1$, $\boldsymbol{x}_{k+1}$) = corr($\boldsymbol{x}_1$, $\boldsymbol{x}_{k+2}$) = corr($\boldsymbol{x}_2$, $\boldsymbol{x}_{k+3}$) = · · · = corr($\boldsymbol{x}_k$, $\boldsymbol{x}_{3k}$) = 0.5. Note that covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_k$ are “low noise” perturbations of the latent variables and constitute our “target covariates”. Variables $\boldsymbol{x}_{3k+1}, ..., \boldsymbol{x}_p$ are independent noise covariates and variables $\boldsymbol{x}_{k+1}, ..., \boldsymbol{x}_{3k}$ are noise covariates that are correlated with the target covariates. The number of observations is set to $n = 150$.

The case of moderate high-dimensional data is represented by the second configuration. We generate $n = 100$ observations from a p-dimensional normal distribution N(0, Σ), with $p = 250$. The covariance matrix $\Sigma = (\Sigma_{ij})\ 1 \leq i,j \leq p$ is given by $\Sigma_{ij} = 0.5^{|i-j|}$, creating correlated predictor variables. Using the coefficient vector $\beta = (\beta_j)\ 1 \leq j \leq p$ with $\beta_1 = \beta_7 = 1.5$, $\beta_2 = 0.5$, $\beta_4 = \beta_{11} = 1$, and $\beta_j = 0$ for $j \in \{1, . . . , p\} \backslash \{1, 2, 4, 7, 11\}$, the response variable is generated according to the regression model (2.7), where the error terms follow a normal distribution with $\sigma = 0.5$.

Finally, the third configuration covers the case of high dimensional data with $n = 100$ observations and $p = 500$ variables. The first 250 predictor variables are generated from a multivariate normal distribution N(0, Σ) with $\Sigma_{ij} = 0.6^{|i-j|}$. Furthermore, the remaining 250 covariates are standard normal variables. Then the response variable is generated according to (2.7), where the coefficient vector $\beta = (\beta_j)$ $1 \leq j \leq p$ is given by $\beta_j = 1$ for $1 \leq j \leq 10$ and $\beta_j = 0$ for $11 \leq j \leq p$, and the error terms follow a standard normal distribution.

To allow for a fraction of outliers we considered the following sampling distributions, listed in increasing order of difficulty

1. No contamination.
2. Vertical outliers: 10% of the error terms in the regression model follow a normal N(20, σ) instead of a N(0, σ).
3. Leverage points: Same as in 2, but the 10% contaminated observations contain high-leverage values by drawing the predictor variables from independent N(50,1) distributions.
4. The outliers form a dense cluster: Keeping the contamination level at 10%, outliers in the predictor variables are drawn from independent N(10, 0.01) distributions. Let $\tilde{\boldsymbol{x}}_i$ denote such a leverage point. Then the values of the response variable of the contaminated observations are generated by $\tilde{\boldsymbol{y}}_i = \eta \tilde{\boldsymbol{x}}_i' \gamma$ with $\gamma = (-1/p)$ $_{1 \leq j \leq p}$. The parameter η controls the magnitude of the leverage effect and is varied from 1 to 25 in five equidistant steps.

5.3.2 Our contribution in SimFrame: Data generation and contamination

We extended model-based data generation for the three configurations by implementing the function (myDataGeneration) and using it as the distribution slot in the DataControl object. This function should have the number of observations to be generated as its first argument. See Code skeleton for a user-defined data generation method in Figure 5.4.

Control class DCARContControl supports specifying a distribution function for generating the contamination. DCARContControl extends ContControl by distribution function for generating the data for the contamination. We also implemented this function to contaminate the data models. See the code skeleton in Figure 5.5.

5.3.3 Simulation Results

In this subsection, the results for the different data scheme are presented and discussed. The performance of the estimated models are compared by the *root mean squared* *prediction error* (RMSPE). For this purpose, we generate *n* additional observations from the respective sampling schemes (without outliers) as test data, and this in each simulation run. The RMSPE of the oracle estimator, which uses the true coefficient values *β*, is computed as a standard for the evaluated methods. In addition considering sparsity, the estimated models are evaluated by the *false positive rate* (FPR) and the *false negative rate* (FNR). Both FPR and FNR should be as small as possible for a sparse estimator. RMSPE, FPR and FNR, averaged over 100 simulation runs, are reported for every method.

5.3.3.1 The First Sampling Scheme

The simulation results for the first data are represented in Table 5.1. It can be seen that when there is no contamination in the data LAD-Lasso, RLARS and Lasso have excellent performance in RMSPE and FPR, while sparse LTS and MM-Lasso have a good prediction, but they have larger FPR than other methods. In addition, MM-Lasso improves the estimates of sparse LTS, which is reflected in the lower values for RMSPE and FPR. On the other hand, there are no false negatives in all of these methods.

In the case of vertical outliers, the higher values of RMSPE and FPR show that Lasso is non-robust estimator. All of methods are still have excellent performance in RMSPE but sparse LTS and MM-Lasso have considerable values of FPR. As showed in Table 5.1 RMSPE and FPR of MM-Lasso are 1.1765, 0.237 while sparse LTS have RMSPE and FPR equals 1.2378, 0.293 respectively. Ultimately, MM-Lasso has a significant improvement over Sparse LTS. In the third scenario, when we introduce leverage points in addition to vertical outliers, RLARS, MM-Lasso, and sparse LTS have a good performance. However, the RMSPE and FPR of RLARS increased (1.1210 to 1.2236, and 0.029 to 0.126, respectively) also the FPR of sparse LTS (0.293 to 0.319) and MM-Lasso (0.237 to 0.250) slightly increase. MM-Lasso still improves the performance of sparse LTS in RMSPE and FPR (1.1792 and 0.250 respectively). LAD-lasso has large RMSPE and suffers from false positives, while Lasso has large RMSPE and FNR. This suggests that the leverage points have a bad leverage effect.

Figure 5.6 refers to the results for the fourth contamination setting. The RMSPE for the more robust methods is plotted as a function of the parameter η. RLARS has a considerably higher RMSPE than MM-Lasso and sparse LTS for lower values of η, but the RMSPE gradually decreases with increasing η. The RMSPE of sparse LTS in beginning slightly increased then decreased in the next steps. However, MM-Lasso has the lowest RMSPE; thus, their overall performance is the best.

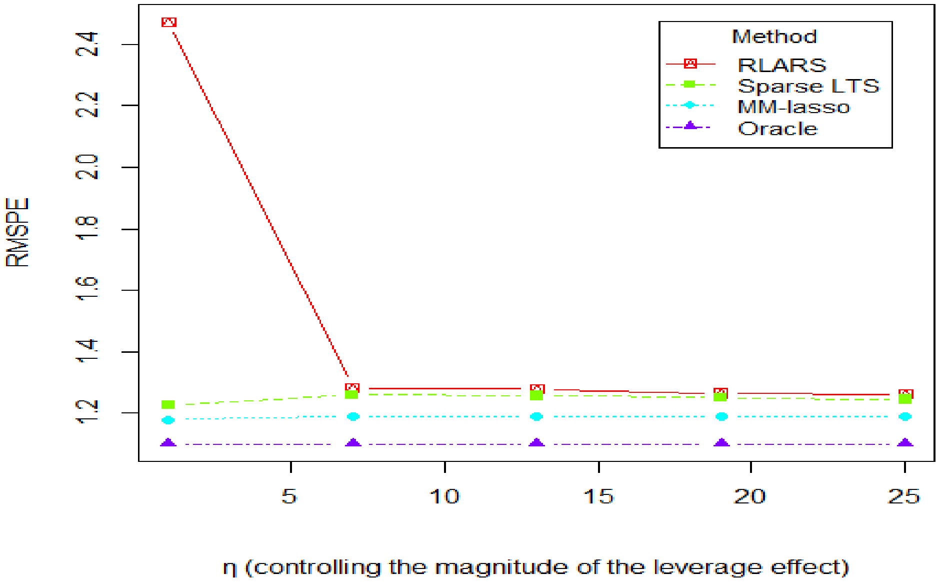

Figure 5.6 Root mean squared prediction error (RMSPE) for the first simulation scheme, with n =150 and p = 50, and for the fourth contamination setting.

Table 5.1 Results for the first simulation scheme, with n = 150 and p = 50.

Method	**No contamination**			**Vertical outliers**			**Leverage points**		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	1.1778	0.080	0.000	1.7376	0.225	0.088	2.5205	0.066	0.766
LAD-Lasso	1.1316	0.092	0.000	1.1640	0.161	0.000	1.9939	0.316	0.002
RLARS	1.1450	0.066	0.000	1.1210	0.029	0.000	1.2236	0.126	0.030
Sparse LTS	1.2623	0.265	0.000	1.2378	0.293	0.000	1.2345	0.319	0.000
MM-Lasso*	1.1705	0.213	0.000	1.1765	0.237	0.000	1.1792	0.250	0.000
Oracle	1.1073			1.1073			1.1073		
* Proposed method.									

5.3.3.2 The second Sampling Scheme

Table 5.2 shows the simulation results for the second data configuration (the moderate high-dimensional data). In the case without contamination, MM-Lasso, and RLARS have the best performance. Also, the LAD-Lasso and Lasso have excellent prediction performance but a slightly higher FPR than the other methods, followed by sparse LTS. In the case of vertical outliers, RLARS still has excellent prediction performance despite some false negatives. We notice that RMSPE and FPR of MM-Lasso are 0.5766 and 0.034 respectively. While, for Sparse LTS are 0.6688, 0.039 respectively. Hence, MM-Lasso achieves good sparse prediction without false negative. Drastically, Lasso is non-robust against vertical outliers. In the scenario with additional leverage points, it can be concluded that sparse LTS has RMSPE equal 0.6691 and FPR equal 0.039 also MM-Lasso has RMSPE equal 0.5738 and FPR equal 0.035. It means that there is stability in these methods. For RLARS, there is small increase in the RMSPE, FPR and FNR. On the other hand, LAD-Lasso already has a considerably large RMSPE, and again a slightly higher FPR than the other methods. Furthermore, the Lasso is still highly influenced by the outliers, which is reflected in a very high FNR and poor prediction performance. Briefly, compared to other methods MM-Lasso is deemed a best performance.

Figure 5.7 clarifies the results for the fourth contamination setting. As in first scheme, we plotted the RMSPE for the more robust methods. The RMSPE of RLARS is gradually decreasing. The RMSPE of MM-Lasso and sparse LTS have constant low values. MM-Lasso clearly performs best for all values of η.

Table 5.2 Results for the second simulation scheme, with n = 100 and p =250.

Method	**No contamination**			**Vertical outliers**			**Leverage points**		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	0.5848	0.105	0.000	2.3551	0.185	0.092	2.6857	0.013	0.632
LAD-Lasso	0.6020	0.067	0.000	0.7446	0.011	0.000	1.8398	0.096	0.112
RLARS	0.5506	0.016	0.000	0.6092	0.015	0.055	0.7901	0.072	0.098
Sparse LTS	0.7195	0.028	0.000	0.6688	0.039	0.000	0.6691	0.039	0.000
MM-Lasso*	0.5526	0.022	0.000	0.5766	0.034	0.000	0.5738	0.035	0.000
Oracle	0.4998			0.4998			0.4998		
* Proposed method.									

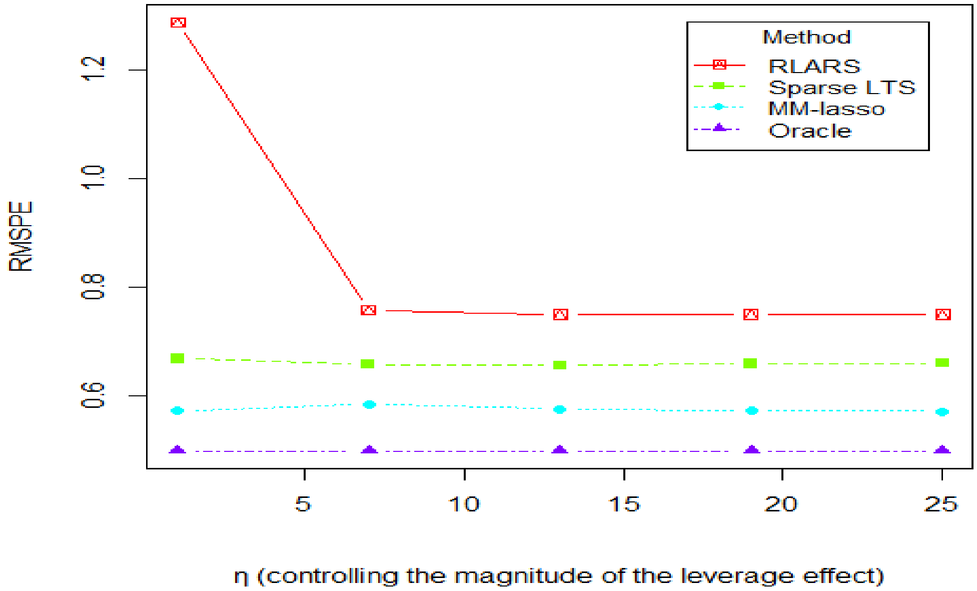

Figure 5.7 Root mean squared prediction error (RMSPE) for the second simulation scheme, with n =100 and p = 250, and for the fourth contamination setting.

5.3.3.3 The Third Sampling Scheme

Table 5.3 presents the simulation results for the high dimensional data configuration. When the data is free from contamination, the sparse LTS is characterized as the lowest efficiency due to have larger values of RMSPE than other methods. In the other hand, MM-Lasso and RLARS have considerably better performance in this case. Lasso and LAD-Lasso have a good behavior. With vertical outliers, the RMSPE for the Lasso increases extremely due to a high FNR, while LAD-Lasso still has good prediction performance. In addition, RLARS has a larger FNR, resulting in a slightly lower RMSPE. When leverage points are introduced, MM-Lasso exhibits the lowest RMSPE and sparse LTS keep its excellent behavior.

Figure 5.8 shows the results for the fourth contamination setting. It can be seen that RMSPE of RLARS is higher in the beginning, and then decreases continuously in the remaining steps. MM-Lasso and Sparse LTS have low and constant values for RMSPE but MM-Lasso is close to Oracle.

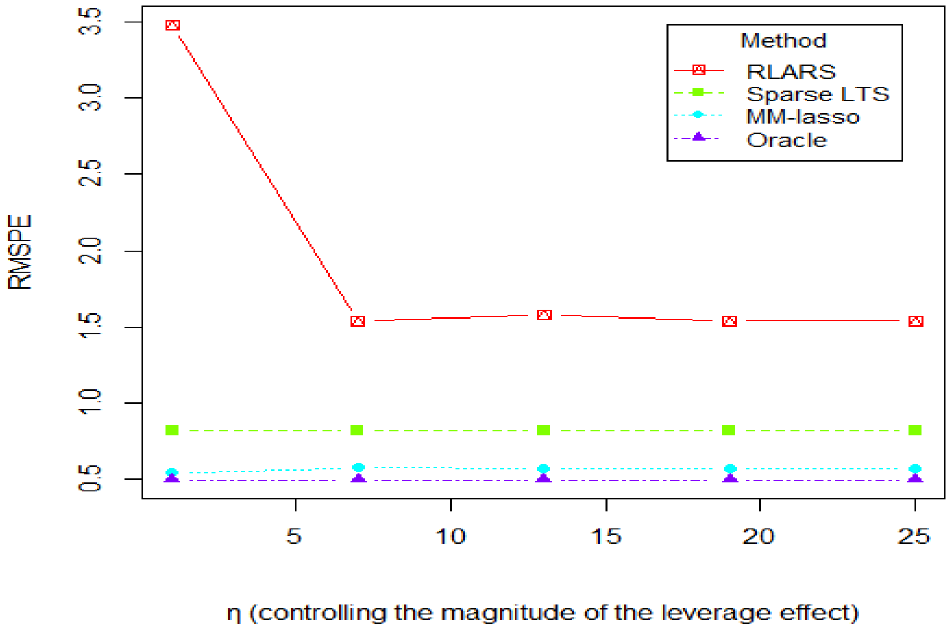

Figure 5.8 Root mean squared prediction error (RMSPE) for the third simulation scheme, with n =100 and p =500, and for the fourth contamination setting.

5.3.3.4 Summary of simulation results

This study shows that MM-Lasso has RMSPE values close to Oracle, and does not suffer from any false positives at all. Hence, MM-Lasso is the best overall performance.

Sparse LTS generally have good prediction accuracy; however, MM-Lasso can improve this prediction. Although RLARS has good achievement, contamination data makes FNR values increased. These simulation results also enhance that the Lasso is not robust to outliers and LAD-Lasso is not robust against bad leverage points but is resistant to vertical outliers.

Table 5.3 Results for the third simulation scheme, with n =100 and p = 500.

Method	**No contamination**			**Vertical outliers**			**Leverage points**		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	0.6114	0.073	0.000	3.3311	0.061	0.280	3.6073	0.054	0.350
LAD-Lasso	0.6493	0.023	0.000	0.8316	0.006	0.000	2.8667	0.074	0.132
RLARS	0.5767	0.009	0.000	0.8954	0.009	0.081	1.4805	0.052	0.112
Sparse LTS	0.9804	0.006	0.000	0.7912	0.005	0.000	0.7520	0.005	0.001
MM-Lasso*	0.5429	0.005	0.000	0.5626	0.004	0.000	0.5725	0.005	0.000
Oracle	0.4983			0.4983			0.4983		
* Proposed method.									

CHAPTER 6

CONCLUSION AND DISCUSSION

We conclude the main results obtained of this thesis:

1. The ordinary least squares (OLS) method, which is widely used for regression modeling, is not appropriate for high dimensional data and statistical estimation is fundamentally different.
2. Sparse linear regression usually describes feature and variable selection problems in high-dimensional linear models.
3. The least absolute shrinkage and selection operator (Lasso) can be used in sparsity high-dimensional linear models. It can effectively select important explanatory variables and estimate regression parameters simultaneously. However, Lasso is variable selection consistent under certain conditions, but not in general and it is not robust to outliers.
4. LAD-Lasso can do regression shrinkage and selection like Lasso and is resistant to vertical outliers but not robust against bad leverage points.
5. Sparse Least trimmed squares (Sparse LTS) is a robust, shrinkage and selection regression estimation with high breakdown value and good prediction estimation.
6. Our method, MM-Lasso can produce robust and sparse linear model for high dimensional data. It can improve prediction estimation of sparse LTS and its overall performance is the best.

We would like to point out some of future research based on our achievement work in this thesis.

1. From simulation study, it can be noted that the models still contain a large number of variables given the small number of observation. Therefore, it may be useful to improve our estimator for producing a sparser model with equal or lower prediction. There are more than approach can be added to MM-Lasso to obtain more beneficial estimator for large p and this is considered for future work.

   a) Relaxed Lasso, proposed by [113], is a relaxation of the penalty for the selected variables of an initial lasso fit. The relaxed Lasso can produce a sparser model with equal or lower prediction loss than the regular Lasso estimator for high dimensional data.

   b) Adaptive Lasso is an approach proposed by [65] to obtain a convex objective function, which yields oracle estimators by using a weighted L_1 penalty with weights determined by an initial estimator.

2. As any study, also this one is subject to some limitations. One limitation is that the values of the tuning constants in the loss functions of the M-estimators were selected to achieve a given efficiency in the non penalized case. One could imagine to select the λ parameter simultaneously with the other tuning constants.

3. We did not provide any asymptotics for MM-Lasso, as was for penalized M-estimators, studied by [114]. Deriving the asymptotic distributions of MM-Lasso for fixed penalty parameter λ seems feasible (but not trivial) if n $n \to \infty$ with p fixed. However, in practice λ is often chosen with a data-driven approach. In addition, the result would be irrelevant for high dimensional data, i.e. with $\mathrm{p} \to \infty$ faster than n i.e. with $\mathrm{p}_{\mathrm{n}}/\mathrm{n} \to \infty$ as $\mathrm{n} \to \infty$, in this case there would interested to investigate in asymptotics theory for MM-Lasso.

4. Solving the problem of asymptotic distribution can help us to estimate the covariance matrix $\boldsymbol{\Sigma}$ of MM-Lasso for high-dimensional datasets. It is challenging because:

   (i) When p > n there are $p(p+1)/2$ unknown parameters to estimate from observations n;

   (ii) $\boldsymbol{\Sigma}$ is intrinsically positive definite.

Estimation of the covariance matrix for high-dimensional datasets is attracting increasing recent attention [115, 116] and investigation of this subject is left for further research.

REFERENCES

[1] Hoerl, A.E. and Kennard, R.W., (1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”, Technometrics, 12(1):55–67.

[2] Tibshirani, R., (1996). “Regression shrinkage and selection via the lasso,” J. Royal. Statist. Soc B., 58(1):267–288.

[3] Knight, K. and Fu, W., (2000). “Asymptotics for Lasso-Type Estimators,” The Annals of Statistics, 28:1356–1378.

[4] Zhao, P. and Yu, B., (2006). “On model selection consistency of LASSO”, Journal of Machine Learning Research, 7:2541–2563.

[5] Zou, H., Hastie, T. and Tibshirani, R., (2007). “On the ‘degrees of freedom’ of the LASSO”, The Annals of Statistics, 35(5): 2173–2192.

[6] Leng, C., Lin, Y. and Wahba, G., (2006). “A note on the Lasso and related procedures in model selection”, Statistica Sinica, 16:1273–1284.

[7] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R., (2004). “Least Angle Regression (with discussion)”, The Annals of Statistics, 32(2):407–499.

[8] Bunea, F., Tsybakov, A. and Wegkamp, M., (2007). “Sparsity orcale inequalities for the LASSO”, Electronic Journal of Statistics, 1:169–194.

[9] Bickel, P. J., Ritov Y. and Tsybakov, A. B., (2009) “Simultaneous analysis of Lasso and Dantzig selector”, The Annals of Statistics, 37(4):1705–1732.

[10] Akaike, H., (1973), “Information theory and an extension of the maximum likelihood principle. In Second International Symposium on Information Theory (B. N. Petrov and F. Cs´aki, eds.), Academiai Kiado, Budapest, 267–281.

[11] Schwarz, G., (1978). “Estimating the dimension of a model”, Ann. Statist, 6:461–464.

[12] Mallows, C., (1973). “Some comments on C_P”. Technometrics, 15: 661–675.

[13] Efron, B., Hastie T. and Tibshirani, R., (2004), “Least angle regression,” The Annals of Statistics, 32:407–499.

[14] Yuan, M. and Lin, Y., (2006). “Model selection and estimation in regression with grouped variables”, Journal of the Royal Statistical Society, Series B 68:49–67.

[15] Gertheiss, J. and Tutz, G., (2010). “ Sparse modeling of categorical explanatory variables”, The Annals of Applied Statistics, 4:2150–2180.

[16] Zou, H. and Hastie, T., (2005). “Regularization and variable selection via the

elastic net", Journal of the Royal Statistical Society B, 67:301–320.

[17] Alfons, A., Croux, C. and Gelper, S., (2013). "Sparse least trimmed squares regression for analyzing high dimensional large data sets", The Annals of Applied Statistics, 7(1):226–248.

[18] Wang H., Li G., and Jiang G., (2007). "Robust regression shrinkage and consistent variable selection through the LAD-lasso", Journal of Business & Economic Statistics, 25:347-355.

[19] Rosset, S. and Zhu, J., (2007). "Piecewise linear regularized solution paths", The Annals of Statistics, 35(3):1012–1030.

[20] Arslan, O., (2012). "Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression", Computational Statistics and Data Analysis, 56:1952– 1965.

[21] Peng, G., Li, H. and Zhu, L., (2013). "Nonconcave penalized M-estimation with a diverging number of parameters", Statitica Sinica, 21(1):391–419.

[22] Maronna, R. A., (2011). "Robust ridge regression for high-dimensional data", Technometrics, 53:44–53.

[23] Aelst, J. A., Van, S. and Zamar, R.H, (2007). "Robust linear model selection based on least angle regression", Journal of the Statistical Association, 102:1289–1299.

[24] Rousseeuw, P.J. and Leroy, A.M., (1987). Robust regression and outlier detection, John Wiley & Sons.

[25] Yohai, V. J. (1987). "High Breakdown-point and High Efficiency Estimates for Regression", The Annals of Statistics, 15: 642-65.

[26] Hubert, M. and Rousseeuw, P.J., (1997)."Robust regression with both continuous and binary regressors", Journal of Statistical Planning and Inference, 57:153–163.

[27] Draper, N.R. and Smith, H., (1998). Applied Regression Analysis (Third Edition). New York: Wiley.

[28] Rousseeuw, P.J. and Leroy, A.M, (2003). Robust Regression and Outlier Detection, New York: Wiley.

[29] Verardi, V., Croux, C., (2009). "Robust regression in Stata", The Stata Journal, 3:439–453.

[30] Huber, P.J., Ronchetti E.M., (2009) Robust Statistics, 2nd ed.; Wiley: New York, NY, USA.

[31] Andersen, R., (2008). Modern Methods For Robust Regression, Thousand Oaks: SAGE Publications.

[32] Hampel, F.R., (1968). Contributions to the Theory of Robust Estimation, Ph.D. Thesis, University of California, Berkeley.

[33] Hampel, F.R., (1974). "The influence curve and its role in robust estimation", J. Amer. Statist. Assoc, 69:383–393.

[34] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J. and Stahel,W.A., (1986). Robust Statistics: The Approach Based on Influence Functions. John Wiley &

Sons :New York, NY, USA.

[35] P. J. Huber, Robust Statistics. Wiley Series in Probability and Mathematical Statistics,New York, New York: John Wiley & Sons, 1981.

[36] Fernholz, L.T., (1983). Von Mises Calculus for Statistical Functionals. Springer, New York.

[37] Boos, D.D. and Serfling, R.J., (1980) . “A note on differentials and CLT and LIL for statistical functions, with application to M-estimates”, The Annals of Statistics, 8:618-624.

[38] Huber, P.J. (1981). Robust Statistics. Wiley, New York.

[39] Edgeworth, F.Y. (1887), “On Observations Relating to Several Quantities”, Hermathena, 6:279-285.

[40] Huber, P. J. (1973). “Robust regression: Asymptotics, conjectures, and Monte Carlo”, Ann. Math. Statist. 1, 799–821.

[41] Rousseeuw, P. J. (1984). “Least median of squares regression”, J. Amer. Statist. Assoc.79:871–880.

[42] Mosteller, F. and Tukey, J.W., (1977). Data Analysis and Regression. Addison–Wesley, New York.

[43] Huber, P. J. (1964) . “Robust estimation of a location parameter. Ann. Math. Statist”, 35:73–101.

[44] Kim, T. and Muller, C., (2007) “Two stage Huber estimation”, Journal of statistical planning and inference, 405–418.

[45] Rousseeuw, P.J. and Croux, C., (1993). “Alternatives to the median absolute deviation”, J. Am. Stat. Assoc., 88(424):1273– 283.

[46] Van der Vaart , A.W., Asympotic statistics. Cambridge Series in Statistical and Probabilistic Mathematics, 3. Cambridge, 1998.

[47] Beaton, A.E. and Tukey, J.W., (1974). “The fitting of power series, meaning polynomials, illustrated on band- pectroscopic data,” Technometrics, 16:147-185.

[48] Fox, J. and Weisberg, S., (2010). Robust Regression in R.

http://socserv.mcmaster.ca/jfox/Books/Companion/appendix/Appendix-`RobustRegression.pdf

[49] R. Maronna, D. Martin, and V. Yohai, Robust Statistics. John Wiley & Sons, Chichester. ISBN 978-0-470-01092-1, 2006.

[50] Godambe, V.P., (1960) “An Optimum Property of Regular Maximum Likelihood Estimation,” Ann. Math. Statist. 31(4):1208-1211.

[51] Holland, P.W., and Welsch, R. E. (1977). “Robust regression using iteratively reweighted least squares”, Commun. Statist. (Theory and Methods), 6:813–828.

[52] Víšek, J. Á.,(2000). “On the diversity of estimates”, Computational Statistics and Data Analysis, 34: 67 - 89.

[53] Kim, J., and Pollard, D., (1990), “Cube Root Asymptotics”, The Annals of Statistics, 18:191–219.

[54] Rousseeuw, P.J. and Van Driessen, K., (1999). "A fast algorithm for the minimum covariance determinant estimator", Technometrics, 41:212–223.

[55] Rousseeuw, P.J. and Van Driessen, K., (1998), "Computing LTS Regression for Large Data Sets",Technical Report, University of Antwerp, submitted.

[56] Rousseeuw, P.J. and Yohai, V.J., (1984), "Robust Regression by Means of S Estimators in *Robust and Nonlinear Time Series Analysis* ", ed. J. Franke, W. Härdle, and R.D., Martin, Lecture Notes in Statistics, 26, New York: Springer-Verlag, 256–274.

[57] Craven, P. and Wahba G., (1979). "Smoothing noisy data with spline functions", Numerische Mathematik, 31:377–403.

[58] Hastie, T., Tibshirani, R. and Friedman, J., (2001). The Elements of Statistical Learning; Data Mining, Inference and Prediction. Springer, New York.

[59] Tibshirani, R.J. and Tibshirani, R., (2009). "A bias correction for the minimum error rate in cross-validation", Annals of Applied Statistics, 3:822–829.

[60] Stein, C., (1981). "Estimation of the mean of a multivariate normal distribution", Ann. Statist, 9:1135–1151.

[61] Ye, J., (1998). "On measuring and correcting the effects of data mining and model selection", J. Amer. Statist. Assoc, 93:120–131.

[62] Hastie, T. and Tibshirani, R., (1990). Generalized Additive Models. Chapman and Hall, London.

[63] Shen, X. and Ye, J., (2002). "Adaptive model selection". J. Amer. Statist. Assoc. 97:210–221.

[64] Efron, B., (2004). "The estimation of prediction error: Covariance penalties and crossvalidation (with discussion)", J. Amer. Statist. Assoc, 99:619–642.

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

[66] Rosset, S. and Zhu, J., (2007). "Piecewise Linear Regularized Solution Paths", The Annals of Statistics, 35(3):1012–1030.

[67] Buhlmann, P., (2011). Statistics for High-Dimensional Data*:* Methods, Theory and Applications , Springer Series in Statistics

[68] Friedman, J., Hastie, T., Hoefling, H.and Tibshirani, R., (2007). "Pathwise Coordinate Optimization", The Annals of Applied Statistics, 2(1):302–332.

[69] Fu, W.J., (1998). "Penalized regressions: The bridge versus the lasso", J. Comput. Graph. Statist, 7:397–416.

[70] Daubechies, I., Defrise, M. and Mol, C.D., (2004). "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", Comm. Pure Appl. Math, 57:1413–1457.

[71] Wu, T.T. and Lange, K., (2008). Supplement to "Coordinate descent algorithms for lasso penalized regression."

[72] Tseng, P., (2001)."Convergence of block coordinate descent method for nondifferentiable maximization", J. Optim. Theory Appl, 109:473–492.

[73] Tseng, P. and Yun, S., (2009). "A coordinate gradient descent method for

nonsmooth separable minimization", Mathematical Programming, 117:387–423.

[74] Lange, K., (2004). Optimization, Springer, New York.

[75] Van de Geer, S., (2007). The deterministic Lasso. In JSM proceedings, (see also http://stat.ethz.ch/research/research reports/2007/140). American Statistical Association.

[76] Van de Geer, S.A. and Bühlmann, P., (2009). "On the conditions used to prove oracle results for the Lasso", Electron. J. Statist, 3:1360-1392.

[77] Donoho, D.L. and Johnstone, I.M., (1994). "Ideal Spatial Adaptation by Wavelet Shrinkage", Biometrika, 81:425-55.

[78] Meinshausen, N. and Bühlmann, P., (2006). "High dimensional graphs and variable selection with the lasso", Ann. Statist, 34:1436–1462.

[79] Zou, H. and Hastie, T., (2005). "Regularization and variable selection via the elastic net", Journal of the Royal Statistical Society B, 67:301–320.

[80] Gao, X. and Huang, J., (2010). "Asymptotic analysis of high-dimensional lad regression with lasso", Statistica Sinica, 20:1485-1506.

[81] Khan, J., Van Aelst, S., and Zamar, R. (2007b). "Robust linear model selection based on least angle regression", Journal of the American Statistical Association, 102(480):1289–1299.

[82] Maronna, R.A., (1976). "Robust M-estimators of multivariate location and scatter", The Annals of Statistics, 4:51-67.

[83] Stahel, W.A. (1981). Breakdown of covariance estimators. Research Rept. No. 31, E.T.H., Zurich.

[84] Alqallaf, F. A., Konis, K.P., Martin, R.D. and Zamar, R. H., (2002). "Scalable robust covariance and correlation estimates for data mining". Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, 14–23.

[85] Maronna, R.A. and Yohai, V.J., (2010). "Correcting MM Estimates for Fat Data Sets", Computational Statistics & Data Analysis, 54:3168-3173.

[86] Yohai, V.J. and Zamar, R.H., (1988). "High breakdown-point estimates of regression by means of the minimization of an efficient scale" Journal of the American Statistical Association, 83:406–413.

[87] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. U Michigan Press.

[88] John, H. (1992). Holland, Adaptation in Natural and Artificial Systems., MIT Press, Cambridge, MA.

[89] Bathe, K., Wilson, E.L. and Peterson, F.E., (1974). SAP IV: A Structural Analysis Program for Static and Dynamic Response of Linear Systems. 73: College of Engineering, University of California Berkeley.

[90] Kostami, V. and Ward, A.R., (2010). Analysis and Comparison of Inventory Systems: Dynamic vs Static Policies, 1–40.

[91] Roacnbluth, A. and Wiener, N., (1945). ”The role of models in science”, Philos. Sci., XII, 4:316-321.

[92] Bettonvil, B. and Kleijnen, J.P., (1997). “Searching for Important Factors in Simulation Models with many Factors: Sequential Bifurcation”, European Journal of Operational Research, 96(1):180–194.

[93] Churchman, C.W., Ackoff, R.L. and Arknoff, E.L., (1959). Introduction to Operatins Research, Wiley, New York.

[94] Kiviat, P. J., (1967). “Digital Computer Simulation: Modeling Concepts”, Report RM-5378-PR, The Rand Corporation, Santa Monica, California.

[95] Rubinstein, R.Y. and Kroese, D.P., (2011). Simulation and the Monte Carlo Method.: Wiley. 372.

[96] Kleijnen, Jack P.C., (1974). “Statistical techniques in simulation part I”, Statistics: textbooks and monographs, Marcel Dekker Inc., New York, 9:285.

[97] Naylor, T.J., Balintfy, J.L., Burdick, D.S. and Chu, K., (1966). Computer Simulation Techniques, Wiley, New York.

[98] Rubinstein, Reuven Y., (1981). “Simulation and the Monte Carlo Method”, John Wiley & Sons, New York-Chichester-BrisbaneToronto, 278 .

[99] Hawkins D.M., (1980). Identification of Outliers, Chapman and Hall.

[100] Barnett, V., and Lewis T., (1994). Outliers in Statistical Data. 3rd edition. J. Wiley & Sons.

[101] Hadi, A.S., (1992). ”Identifying multiple outliers in multivariate data,” Journal of the Royal Statistical Society. Series B, 54:761-771.

[102] Alfons, A., (2012b). simFrame: Simulation framework. R package version 0.5.

Gabriele B. Durrant, Imputation Methods for Handling Item-Nonresponse in the Social Sciences: A Methodological Review, ESRC National Centre for Research Methods,

http://www.napier.ac.uk/depts/fhls/peas/pdfs/durrantreview.pdf (30.09.2005).

[103] Chambers, J., (1998). Programming with Data. Springer-Verlag, New York.

[104] Chambers, J., (2008). Software for Data Analysis: Programming with R. Springer-Verlag, New York.

[105] Fowler, M., (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd edition. Addison-Wesley. ISBN 978-0-321-19368-1.

[106] Genz, A., Bretz, F., (2009). Computation of Multivariate Normal and t Probabilities, volume 195 of Lecture Notes in Statistics. Springer-Verlag, New York.

[107] Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F. and Hothorn, T., (2010). mvtnorm: Multivariate Normal and t Distributions. R package version 0.9-92, URL http://CRAN.R-project. org/package=mvtnorm.

[108] Be´guin C., Hulliger, B., (2008). “The BACON-EEM Algorithm for Multivariate Outlier Detection in Incomplete Survey Data.” Survey Methodology, 34(1):91–103.

[109] Hulliger, B., Schoch T., (2009a). “Robust Multivariate Imputation with Survey Data.” 57th Session of the International Statistical Institute, Durban.

[110] Alfons, A., (2012a). robustHD: Robust methods for high-dimensional R pakage version 0.1.0.

[111] Koenker, R., (2011). quantreg: Quantile regression. R package version 4.67.

[112] Hasti, T. and Efron, B., (2011). lars: Least angle regression, lasso and forward stagewise. R package version 0.9-8.

[113] Meinshausen, N., (2007). ”Relaxed Lasso”, Computational Statistics & Data Analysis 52:374–393.

[114] Germain, J. F. and Roueff, F. (2010). Weak convergence of the regularization path in penalized M-estimation. Scandinavian Journal of Statistics, 37:477–495.

[115] Bickel, P.J. and Levina, E., (2008). “Covariance Regularization by Thresholding”, The Annals of Statistics, 36(6):2577–2604.

[116] Bickel, P.J. and Levina, E., (2008). “Regularized Estimation of Large Covariance Matrices”, The Annals of Statistics, 36(1):199–227.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname	: KAMAL S.A. DARWISH
Date of birth and place	: 19/9/1972 Palestine
Foreign Languages	: English , Turkish
E-mail	: kdarweesh.scom@gmail.com

EDUCATION

Degree	**Department**	**University**	**Date of Graduation**
Master	Scientific Computing	Birzeit University	2006
Undergraduate	Mathematics	Al-Quds University	1997
High School	Science	Gaza secondary school	1991

WORK EXPERIENCE

Year	**Corporation/Institute**
2011	Alquds open university
2008	Department of Computer Science /Birzeit University
2001	Ministry of Education /Palestine

PUBLISHMENTS

Papers

1. Darweesh, K., and Saleh, M., (2007). “Efficient Elliptic Curve Cryptosystems Using Efficient Exponentiation” , eJournal of Engineering Mathematics: Theory and Application, 1:1-15

2. Darwish, K. and Büyüklü, A.H., (2015).” Robust Linear Regression Using L_1-Penalized MM-Estimation for High Dimensional Data” American Journal of Theoretical and Applied Statistics, 4(3):78-84.