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

NEW ROBUST PENALIZED ESTIMATORS FOR LINEAR AND LOGISTIC REGRESSION

FATMA SEVİNÇ KURNAZ

PhD. THESIS
DEPARTMENT OF STATISTICS
PROGRAM OF STATISTICS

ADVISER
ASSOC. PROF. DR. ATIF AHMET EVREN

CO-ADVISER
PROF. DR. PETER FILZMOSER

ISTANBUL, 2017

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

NEW ROBUST PENALIZED ESTIMATORS FOR LINEAR AND LOGISTIC REGRESSION

A thesis submitted by Fatma Sevinç KURNAZ in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 19.06.2017 in Department of Statistics, Statistics Program.

Thesis Adviser

Assoc. Prof. Dr. Atıf Ahmet EVREN

Yıldız Technical University

Co-Adviser

Prof. Dr. Peter FILZMOSER

Vienna University of Technology

Approved By the Examining Committee

Assoc. Prof. Dr. Atıf Ahmet EVREN

Yıldız Technical University _____

Prof. Dr. Ali Hakan BÜYÜKLÜ, Member

Yıldız Technical University _____

Assoc. Prof. Dr. Doğan YILDIZ, Member

Yıldız Technical University _____

Prof. Dr. Müjgan TEZ, Member

Yıldız Technical University _____

Assoc. Prof. Dr. Kadri Ulaş AKAY, Member

Yıldız Technical University _____

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: 2214A.

ACKNOWLEDGEMENTS

I am grateful to the God for the good health and wellbeing that were necessary to complete this thesis.

I would like to thank my advisor Prof. Atıf Ahmet EVREN for giving me the opportunity to start the journey of my PhD in Department of Statistics at Yildiz Technical University. I remain indebted for his understanding and support during the times when I was really down and depressed due to some problems. Your support has been excellent! Thank you!

I would like to express my sincere gratitude to my co-advisor Prof. Peter FILZMOSER for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. Words cannot express how thankful I am for his support in every aspect of my academic development. Without him and his continuous guidance, I never would have seen the end of this study. Thank you!

Special thanks go to Irene HOFFMANN. Our collaborations were very inspiring and motivating. I have learned a lot from you and am very proud of what we have achieved together. Thank you for your each calling and asking me that "Hi Sevinç! Is there anything to discuss?" and also for teaching me the sentence "mai perdere la speranza!"

I would like to thank the rest of my thesis committee for their insightful comments and encouragement. In particular, I would like to express my gratitude towards Prof. Müjgan TEZ who is always ready to help and listen me for any trouble. Your support is always very important for me as it serves as an inspiration to achieve my goals. Thanks! Special thanks go to my friend Erhan ÇENE for advice, comfort and help during the most difficult periods of the PhD. I truly appreciate your kind help and support, thanks!

Last but not the least, I would like to thank my father Ali KURNAZ, my mother Meryem KURNAZ and to my brother Bahri KURNAZ and friends, particularly Sibel YILMAZ and Burcu KAZANCI, for supporting me spiritually throughout writing this thesis and my life in general. My hard-working parents have sacrificed their lives for my brother and myself and provided unconditional love and care. I love them so much, and I would not have made it this far without them. I will forever be thankful to you. Teşekkürler sevgili annem, babam ve kardeşim.

This dissertation is dedicated to my lovely family; Ali, Meryem and Bahri KURNAZ.

Haziran, 2017

Fatma Sevinç KURNAZ

TABLE OF CONTENTS

Page

$bias$	Bias of the current estimator
H	Index subset of size h
H_{el}^{s}	Elemental subset of size 3 or 4 according to current model
H_{opt}	Optimal index subset of size h
$\boldsymbol{\beta}$	Vector of unknown parameter
$\boldsymbol{\beta}_{H_{el}^{s}}$	Raw elastic net LTS estimator calculated on elemental subset
$\hat{\boldsymbol{\beta}}_{enet}$	The elastic net estimator
$\hat{\boldsymbol{\beta}}_{enetLTS}$	Raw elastic net LTS estimator
$\hat{\boldsymbol{\beta}}_{Liu}$	Liu estimator
$\hat{\boldsymbol{\beta}}_{LTS}$	Least trimmed squares estimator
$\hat{\boldsymbol{\beta}}_{LTS-Liu}$	LTS-Liu estimator
$\hat{\boldsymbol{\beta}}_{M}$	M estimator
$\hat{\boldsymbol{\beta}}_{M-Liu}$	M-Liu estimator
$\hat{\boldsymbol{\beta}}_{MM}$	MM estimator
$\hat{\boldsymbol{\beta}}_{MM-Liu}$	MM-Liu estimator
$\hat{\boldsymbol{\beta}}_{ML}$	Maximum likelihood estimator
$\hat{\boldsymbol{\beta}}_{PRM}$	Partial Robust M estimator
$\hat{\boldsymbol{\beta}}_{PRM-Liu}$	PRM-Liu estimator
$\hat{\boldsymbol{\beta}}_{reweighted}$	Reweighted elastic net estimator
$\hat{\boldsymbol{\beta}}_{Ridge}$	Ridge estimator
$\hat{\boldsymbol{\beta}}_{S}$	S estimator
$\hat{\boldsymbol{\beta}}_{sparseLTS}$	The sparse LTS estimator
P_{α}	Penalty term
$\boldsymbol{\varepsilon}$	Error term
Q	Objective function
r_i	Residuals of the current estimator
w_i	Weights
$\mathbf{X}$	Matrix of the predictors with size of $n \times p$, whose columns indicate variables
$\mathbf{y}$	Predictand vector with size $n \times 1$

BY	Bianco-Yohai Estimator
CV	Cross–Validation
RMSPE	Root Mean Squared Prediction Error
FPR	False Positive Rate
FNR	False Negative Rate
LS	Least Squares
LTS	Least Trimmed Squares
MCR	Misclassification Rate
ML	Maximum Likelihood
MNLL	Mean of Negative Log-Likelihood
MSE	Mean Squared Error
PLS	Partial Least Squares
PRECISION	The Precision of the Coefficient Estimate

LIST OF FIGURES

Page

LIST OF TABLES

Page

NEW ROBUST PENALIZED ESTIMATORS FOR LİNEAR AND LOGİSTİC REGRESSION

Fatma Sevinç KURNAZ

Department of Statistics
PhD. Thesis

Adviser: Assoc. Prof. Dr. Atıf Ahmet EVREN
Co-adviser: Prof. Dr. Peter FILZMOSER

The least squares (LS) regression estimator can be very sensitive in the presence of multicollinearity among predictors and outliers in the data. As a solution, we introduce a new robust version of Liu estimator. Although the proposed estimator is useful for low dimensional data, there are some restrictions of it for high-dimensional data, namely some calculation problems. Respecting this situation, a new robust Liu-type estimator with similar idea is introduced for high-dimensional data. By considering weights, also the resulting estimators are highly robust, but also the estimations of the biasing parameters are robustified.

The main focus of this thesis is to provide a family to literature which is able to deal with multicollinearity among predictors and outliers in the data, particularly high-dimensional data. Concerning improving interpretibility and increasing the model predictive ability in high-dimensional data, variable selection has attracted much research interest. Modern regularization methods have become a popular choice because they perform intrinsic variable selection and parameter estimation simultaneously. However, the estimation procedure becomes more difficult and challenging task when the data suffer from outliers. As a solution, recently, researchers started to improve robust versions of those regualarization methods. With this aim, fully robust versions of the elastic net estimator are introduced for linear regression. Conserning the binary response case, the idea is extended for logistic regression. The algorithms to compute the newly proposed estimators are based on the idea of repeatedly applying the non-robust classical estimators to data subsets only. It is shown how outlier-free subsets can be identified efficiently, and how appropriate tuning parameters for the elastic net penalties can be selected for corresponding model. A final

reweighting steps are thought to improve the efficiency of the estimators.

Simulation studies compare with non-robust and other competing robust estimators and reveal the superiority of the newly proposed methods. This is also supported by a reasonable computation time. Additionaly, some real data examples show the advantages of the proposed estimators.

Keywords: Elastic net penalty, Least trimmed squares, C-step algorithm, High dimensional data, Robustness, Sparse estimation, Liu Estimator.

LİNEER VE LOJİSTİK REGRESYON İÇİN YENİ CEZALI ROBUST TAHMİN EDİCİLER

Fatma Sevinç KURNAZ

İstatistik Bolumu
Doktora Tezi

Danışman: Doç. Dr. Atıf Ahmet EVREN
Eş Danışman: Prof. Dr. Peter FILZMOSER

Veri kümesi sapan değerler içerdiğinde ve açıklayıcı değişkenler arasında çoklu iç ilişki bulunduğunda, En Küçük Kareler (EKK) tahmin edicisi çok hassas olabilmektedir. Çözüm olarak, Liu tahmin edicisinin yeni bir robust (dirençli, sağlam) versiyonunu takdim etmekteyiz. Önerilen bu tahmin edici küçük boyutlu veri kümeleri için kullanışlı olmasına rağmen, çok boyutlu veri kümeleri için bazı sınırlamalara, yani bazı hesaplama problemlerine, sahiptir. Bu durumu göz önüne alarak, çok boyutlu veri kümeleri için benzer bir fikirle yeni bir robust Liu-tip tahmin edici önermekteyiz. Gözlemler ağırlıklandırılarak, yalnızca elde edilen bu tahmin edicilerin sapan değerlere dirençli olması sağlanmamış, aynı zamanda yanlılık parametrelerinin tahmin edicileri de robust hale getirilmiştir.

Bu tezdeki temel amacımız verilerdeki (özellikle çok boyutlu verilerdeki) sapan değerler ve açıklayıcı değişkenler arasındaki çoklu iç ilişki problemini çözmek için yeni bir tahmin edici ailesini literatüre kazandırmaktır. Çok boyutlu veri kümelerinde modelin tahmin yeteneğini artırmak ve yorumlamayı kolaylaştırmak hususları göz önüne alındığında, değişken seçimi konusu araştırmacıların yoğun ilgisini cezbetmektedir. Modern düzenleme yöntemleri aynı anda hem değişken seçimi hem de parametre tahminine imkan verdiği için tercih edilir hale gelmiştir. Ancak veriler sapan değerlerden zarar gördüğünde, tahmin prosedürü daha zor bir hale gelmektedir. Çözüm olarak, araştırmacılar son zamanlarda bu düzenleme yöntemlerinin robust versiyonlarını geliştirmeye başlamışlardır. Bu amaçla, lineer regresyon için elastik net tahmin edicisinin bütünüyle robust bir versionunu takdim etmekteyiz. Yanıt değişkeninin iki kategorili olduğu durum göz önüne alınarak, önerilen bu yöntem lojistik regresyon için genişletilmiştir. Önerilen yeni tahmin edicileri hesaplamak için verilen algoritmalar, robust olmayan klasik tahmin edicilerin verilerin sadece alt

kümelerine tekrar tekrar uygulanması üzerine inşa edilmiştir. Sapan değerlerden ayıklanmış alt kümelerin nasıl belirlenebileceği ve karşılık gelen model için elastik net cezasına ait yanlılık parametrelerinin nasıl uygun bir şekilde seçilebileceği gösterilmiştir. Son olarak, tahmin edicilerin etkinliğini arttırmak için yeniden ağırlıklandırma adımı kullanılmıştır.

Simulasyon çalışmaları robust olmayan tahmin edicilerle ve alternatif robust tahmin edicilerle, önerilen tahmin edicilerin karşılaştırılmasını yapmaktadır ve önerilen tahmin edicilerin üstünlüğünü ortaya koymaktadır. Bu durum, önerilen tahmin edicinin makul bir hesaplama süresine sahip olduğu gösterilerek de desteklenmiştir. Ek olarak, bazı gerçek veri kümeleri üzerinde önerilen tahmin edicilerin avantajları gösterilmektedir.

Anahtar Kelimeler: Elastik net cezası, En küçük kırpılmış kareler, C-adımlar algoritması, Yüksek boyutlu veri kümeleri, Dirençlilik, Sparse tahmin edici, Liu tahmin edici.

CHAPTER 1

INTRODUCTION

1.1 Literature Review

Linear regression is generally designed for low dimensional data sets where the number of observations (n) is greater than the number of predictors (p), and the ordinary least squares (LS) regression is the most common method for linear regression. However, in presence of multicollinearity among predictors, the assumptions of LS regression are violated and the results might give misleading information. As an alternative to LS, the ridge estimator is defined adding an l_2 penalty on the coefficient estimator to the objective function of LS [1]. Another common approach for a biased estimator is the Liu estimator [2]. The idea behind of Liu estimator is to propose an estimator which has similar properties as the ridge estimator, but with an easier calculation of the biasing parameter λ by means of a linear function of λ. This estimator is directly using the LS-estimator, thus it can be affected by outliers in data – observations which are unusually far away from the data center are often referred to as outliers. However, there are two proposals available that are devoted to robustifying the Liu estimator against outliers. The first proposal employs the M-estimator, and the resulting robustified Liu estimator is called as M-Liu estimator, see [3]. Since the M-estimator is not robust against leverage points, the M-Liu estimator has the same drawback. Another robust proposal is the LTS-Liu estimator [4]. Although the LTS-estimator is robust against both leverage points and vertical outliers, there is a question whether this robustness also implies robustness of the LTS-Liu estimator.

On the other hand, high dimensional data sets have also been a current issue due to the growth of improved technology which allows monitoring of thousands of variables.

Analyzing such data sets ($n \ll p$) has become a focus for many researchers in a wide range of scientific fields such as chemometrics, biometrics, econometrics, social sciences and etc. Therefore, the huge demand of resolutions for various statistical problems have emerged in those scientific areas. Particularly in chemometrics, the partial least squares (PLS) regression has become a main tool since years, whose history is closely connected with the history of chemometrics [5, 6]. Another important thing with high dimensional data is that they can include many uninformative variables which have no effect on the predictand or have very small contribution to the model. A regression model including uninformative variables gives unstable results and the interpretation of it is a challenging task. As a solution, sparse estimation methods are proposed to handle the high dimensional data issue. One of the common methods for sparse estimation is the lasso estimator which leads to coefficients of exactly zero [7]. This means the lasso returns a smaller subset of the variables that have highest importance for the model. Therefore lasso can be regarded as variable selection method. The term *sparse* is used for a model with exact zero coefficients. Roughly speaking, adding an l_1 constrait for the coefficient estimates to the objective function yields sparse solutions. The lasso estimator is able to select at most n variables when $n < p$ and this situation has restrictive influence on its variable selection property [7]. Another sparse method, the elastic net, which is the penalized version of LS with both l_1 and l_2 penalties, is introduced by Zou and Hastie [8]. The elastic net estimator is able to select variables like lasso and able to shrink the coefficients according to ridge. For an overview of sparse methods, see [9].

Besides most real world data sets have more variables than observations, which is high dimensional, they also contain outliers that have remarkably large or small values when compared with majority of the data set. Therefore, another important problem in regression analysis are outliers in the data set, which might be only vertical outliers (outliers in the predictand space), only leverage points (outliers in predictors space) or both types. Even though the sparse regression estimation methods are particularly useful for high dimensional data, they are seriously distorted by outliers since they are not robust. To deal with this problem, we need to consider robust statistical techniques. For a wide overview to robust methods, see [10]. As mentioned in low dimensional case, one common and

well studied method is the Least Trimmed Squares (LTS) estimator [11]. Although the LTS estimator has a simple definition and high robustness to outliers, the computational time increases extremely for larger data set. To overcome this problem, the FAST-LTS algorithm was proposed by Rousseeuw [12]. The main idea of the FAST-LTS algorithm is the ”concentration step” or C-step algorithm based on looking for an index subset, which excludes undesired observations by taking the smallest squared residuals. Even if it is very effective method for larger data sets, it does not work due to rank problem of the design matrix when $p > n$ and does not yield a sparse solution. One of the few existing sparse and robust methods is defined adding an l_1 penalty to the objective function of the LTS estimator and is called the sparse least trimmed squares (sparse LTS) regression estimator [13]. The sparse LTS estimator is robust to both vertical outliers and leverage points. Nevertheless, the sparse LTS estimator can be effected by some problems which come from the lasso. For instance, multicollinearity among the predictors may lead to instability of the estimator since it has only l_1 penalty. Another related approaches are based on adding an l_1 penalty to the objective function of MM-estimators [14, 15].

Logistic regression is a standard probabilistic statistical classification model that is widely used in many fields. The main difference to linear regression is that, the response in logistic regression is a binary variable, coding the class-membership of two groups. The most famous method to estimate unknown coefficients is the maximum likelihood (ML) estimator. Similar to linear regression, it is not possible to calculate the ML estimator without modifications when $p > n$. Recently, to solve this problem, Friedman et al. [16] suggested a new estimator using an elastic net penalty, which is obtained by maximizing the penalized binomial log-likelihood function. Whereas the elastic net penalty yields sparse solution thanks to l_1 penalty, the estimated coefficients may be seriously affected by outliers. Several robust but non-sparse alternatives have been proposed in literature [17, 18]. Finding a way to adapt the trimming idea to logistic regression may seem quite attractive to provide a robust method. However, Albert and Anderson [19] proved the ML estimator exists if there is overlap between the observations from each classes. Trimming observations may cause non-overlapping and thus existence problem of the ML estimator. The penalized binomial log-likelihood function overcomes this problem, and lead to a

solution even if there is no overlapping [16, 20]. One robust and sparse estimator for logistic regression is introduced using weights to reduce the influence of outliers by [21]. Their approach is to perform outlier detection in a principal component analysis (PCA) space, obtain weights based on robust Mahalanobis distances in the PCA score space and derive weights from these distances. These weights are then used to down-weight the negative log-likelihood in the penalized objective function to reduce the influence of outliers. However, it is not guaranteed that outliers can be detected in the PCA score space. An increasing number of uninformative variables will disguise observations deviating from the majority only in few informative variables, but these hidden outlying observations can still distort the model. Therefore, model based outlier detection is highly recommended as proposed in our algorithm.

1.2 Objective of the Thesis

In this study, we focus not only on robust estimation, but also on the multicollinearity problem in both low dimension and high dimension. With this aim, we proposed robust version of Liu estimator using the highly robust and efficient MM-estimator as a plug-in estimator. Secondly, we have also provided another robust Liu estimator for the use with high-dimensional small sample size data. This estimator uses the PRM-estimator, a robustified partial least-squares (PLS) estimator [22], as a plug-in estimator.

Another important contribution of this work can be divided into twofold: A new robust and sparse regression estimator is proposed with combined l_1 and l_2 penalties. This robustified elastic net regression estimator overcomes the limitations of lasso type estimators concerning the low number of variables in the models, and concerning the instability of the estimator in case of high multicollinearity among the predictors [7]. As a second contribution, the idea for the linear regression is extended to logistic regression. The resulting estimator is a robust elastic net version of logistic regression. To provide robustification we use the trimming idea. This idea could cause to overlapping problem. However, using the elastic net penalty also solves the non-existence problem of the estimator in case of non-overlapping groups [19, 16, 20]. Therefore, proposed robust and sparse estimators act

like a variable selection method by returning a smaller subset of variables being relevant for the model, inheriting this property from elastic net penalty.

1.3 Hypothesis

This dissertation aims to address the solutions to main problems in regression analysis such as multicollinearity among predictors and outliers in data. With this goal, we introduce the robust and sparse parameter estimation and therefore variable selection for linear regression. Therefore, the proposed method is quite useful for high dimensonal data. Another contribution of this thesis is that those problems are also taken into consideration in context of logistic regression and thus make a first step to extend the algorithm for generalized methods. The robustness of the estimator is achieved by trimming the penalized log-likelihood function, and using weights. For linear regression, weights are determined in the same way with [13]. As for logistic regression, weights are calculated as proposed in the context of robust logistic regression [17, 18]. These weights can also be applied in a reweighting step which increases the efficiency of the robust elastic net linear and logistic regression estimators.

CHAPTER 2

REVIEW TO THE EXISTING METHODS

This chapter is dedicated to give an overview of the existing methods in linear regression and logistic regression in the context of low and high dimensional data. But before going there, we would like to mention some important charactarizations in terms of robustness, such as breakdown point (BP) and influence function (IF).

Breakdown Point

A common measure allowing us to describe the robustness of an estimator is its breakdown point (BP), which is generally defined as the minimum fraction of outliers that is able to completely distort the estimator, i.e. the estimator yields any arbitrary result [10]. On the other side, a simple and intuitive definition of the BP for finite samples was introduced by [23]. Mathematically, let $\tilde{\boldsymbol{\beta}}$ be an estimator for a data set $Z = \{(\mathbf{x}_i, y_i) : 1 \leq i \leq n\}$. Then, the BP of $\tilde{\boldsymbol{\beta}}$ is the

$$\varepsilon^*(\tilde{\boldsymbol{\beta}}) = \min\{\frac{m}{n} \sup_{\tilde{Z}} \left\|\tilde{\boldsymbol{\beta}}\right\|_2 = \infty\} \tag{2.1}$$

where $\tilde{Z}$ indicates outliers obtained from Z by replacing m of the originally n observations by arbitrary values.

Influence Function

The influence function (IF) is a measure of the asymptotic bias of an estimator when the

assumed model is subject to a small amount of contamination by a point mass distribution, and therefore provides only a local approximation to the act of the estimator [24]. The IF of $\tilde{\boldsymbol{\beta}}$ is defined as

$$\mathrm{IF}_{\tilde{\boldsymbol{\beta}}}(x_0, F) = \lim_{\varepsilon \downarrow 0} \frac{\tilde{\boldsymbol{\beta}}_\infty((1-\varepsilon)F + \varepsilon\delta_{x_0}) - \tilde{\boldsymbol{\beta}}_\infty(F)}{\varepsilon} \tag{2.2}$$

where δ_{x_0} is the point mass at x_0, "$\downarrow$" denotes "limit from right side", ε is a fraction of outliers and $\tilde{\boldsymbol{\beta}}_\infty$ is the asymptotic value of the estimator $\tilde{\boldsymbol{\beta}}$ at distribution F. For more information, see [24, 10].

2.1 Linear Regression

We consider the multiple linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.3}$$

where $\mathbf{y} = (y_1, \ldots, y_n)'$ contains the n observations of the response variable. The information of the p explanatory variables observed for the same n observations is collected in the $n \times (p+1)$ matrix $\mathbf{X}$, where the first column of this matrix consists of ones, taking care of the intercept term. Thus, the rows of $\mathbf{X}$ are $(1, \mathbf{x}_i')$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$ is the i-th observation of the explanatory variables, for $i = 1, \ldots, n$. The unknown regression coefficients are $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ (β_0 is the intercept), and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)'$ is the error term, assumed to have mean vector zero and covariance matrix $\sigma^2\mathbf{I}_n$.

The most common method to estimate the regression coefficients is the ordinary Least Squares estimator (LS), which is defined as

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (r_i(\boldsymbol{\beta}))^2, \tag{2.4}$$

where $r_i(\boldsymbol{\beta}) = y_i - (1, \mathbf{x}_i')\boldsymbol{\beta}$ are the residuals. Although the LS-estimator has many desirable statistical properties as compared to other unbiased linear estimators, it can be affected by outliers in the data, i.e. outliers either in the response or in the explanatory variables (or

both) [10]. Thus, the breakdown point of the LS-estimator is zero.

In our study, we focus not only on robust estimation, but also on the multicollinearity problem. Multicollinearity is that two or more predictor variables in a multiple regression model are highly correlated. In practice, generally the issue of near multicollinearity arises, which means that there is an approximate linear relationship among two or more predictor variables. It is well known that with near multicollinearity the LS-estimator becomes unstable since $\mathbf{X}'\mathbf{X}$ is nearly singular. There are several proposals to deal with this problem.

2.1.1 Ridge Estimator

A first solution was provided by [1] who introduced the ridge regression estimator,

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \arg\min_{\boldsymbol{\beta}} \left(\sum_{i=1}^{n} (r_i(\boldsymbol{\beta}))^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \tag{2.5}$$

with the complexity parameter $\lambda \geq 0$ that needs to be selected appropriately in order to optimize the prediction accuracy. The minimization problem (2.5) leads to the closed-form solution

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{y}. \tag{2.6}$$

The choice $\lambda = 0$ leads to the unbiased LS-estimator, and for $\lambda > 0$ one gets a biased estimator.

2.1.2 Liu Estimator

A different proposal for a biased estimator, the LS-Liu estimator, defined as

$$\hat{\boldsymbol{\beta}}_{\text{Liu}} = (\mathbf{X}'\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{X} + \lambda_{\text{LS}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\text{LS}} \tag{2.7}$$

where $0 < \lambda_{\text{LS}} < 1$ is the biasing parameter [2]. The idea was to propose an estimator which has similar properties as the ridge estimator, but with an easier calculation of the biasing parameter since $\hat{\boldsymbol{\beta}}_{\text{Liu}}$ is a linear function of λ_{LS}. Since this estimator is directly using the LS-estimator, it is not robust against outliers.

Although these methods till here are very useful in case multicollinearity among predictors, they are distorted when data include outliers. In the linear regression setting, outliers may appear in the space of the predictand (so-called vertical outliers), or in the space of the predictor variables (leverage points) [10]. There are two different versions of the leverage points, namely a leverage point is called as "*good* leverage point " if it follows the pattern of the majority, and it is called as "*bad* leverage point" if it is far away from the majority.

If data includes outliers, robust methods are the most appropriate methods to use. The main robust methods are summarized below for low dimensional case:

2.1.3 LTS Estimator

A further limitation of the previously mentioned estimators is their lack of robustness against data outliers. In practice, the presence of outliers in data is quite common, and thus robust statistical methods are frequently used, see, for example [25, 26]. The Least Trimmed Squares (LTS) estimator has been among the first proposals of a regression estimator being fully robust against both types of outliers [11]. It is defined as

$$\hat{\boldsymbol{\beta}}_{\text{LTS}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{h} r^2_{(i)}(\boldsymbol{\beta}), \tag{2.8}$$

where the $r_{(i)}$ are the ordered absolute residuals $|r_{(1)}| \leq |r_{(2)}| \leq \cdots \leq |r_{(n)}|$, and $r_i = y_i - \mathbf{x}_i^T \boldsymbol{\beta}$ [27]. The number h is chosen between $\lfloor (n+p+1)/2 \rfloor$ and n, where $\lfloor a \rfloor$ refers to the largest integer $\leq a$, and it determines the robustness properties of the estimator [27]. The LTS estimator also became popular due to the proposal of a quick algorithm for its computation, the so-called FAST-LTS algorithm [12]. The key feature of this algorithm is the "concentration step" or C-step, which is an efficient way to arrive at outlier-free data

subsets where the LS estimator can be applied. The LTS estimator achieves the maximum breakdown point of 50% for $h = \left[\frac{n+p}{2}\right]$, but it only has a low efficiency (since it is an S estimator).

2.1.4 MM Estimator

One of the several alternative regression estimators that are robust to both types outliers is the MM-estimator of regression. It has high breakdown point of 50% and high efficiency [28]. The MM-regression estimator is based on the M-estimator of regression

$$\hat{\boldsymbol{\beta}}_{\mathrm{M}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}(\boldsymbol{\beta})}\right), \tag{2.9}$$

where ρ is a bounded function, like Tukey's biweight function [10]. Here, $\hat{\sigma}$ is the scale estimator of the residuals. In Figure 2.1 it is observed that the bisquare objective function levels off for $|t| > M$ and the weight function give reduced weights at the tails instead of giving weight one to all observations. The M-estimator is not robust against leverage points [10], and thus the MM-estimator uses as a robust residual scale estimator an M-estimator of scale, which is the solution of the equation

$$\sum_{i=1}^{n} \tilde{\rho}\left(\frac{r_i(\boldsymbol{\beta})}{\sigma(\boldsymbol{\beta})}\right) = \delta, \tag{2.10}$$

with $\tilde{\rho}$ taken e.g. as the bisquare function [10], and the tuning constant δ. Regression estimators with $\hat{\sigma}$ given by (2.10) are called S estimators, and they can be generally defined as

$$\hat{\boldsymbol{\beta}}_{\mathrm{S}} = \arg\min_{\boldsymbol{\beta}} \hat{\sigma}\left(r_1(\boldsymbol{\beta}), \ldots r_n(\boldsymbol{\beta})\right). \tag{2.11}$$

S estimators have high breakdown point but low efficiency [10]. It was shown in [28] that the MM-estimator inherits the breakdown point of the S estimator, but allows for a tunable efficiency.

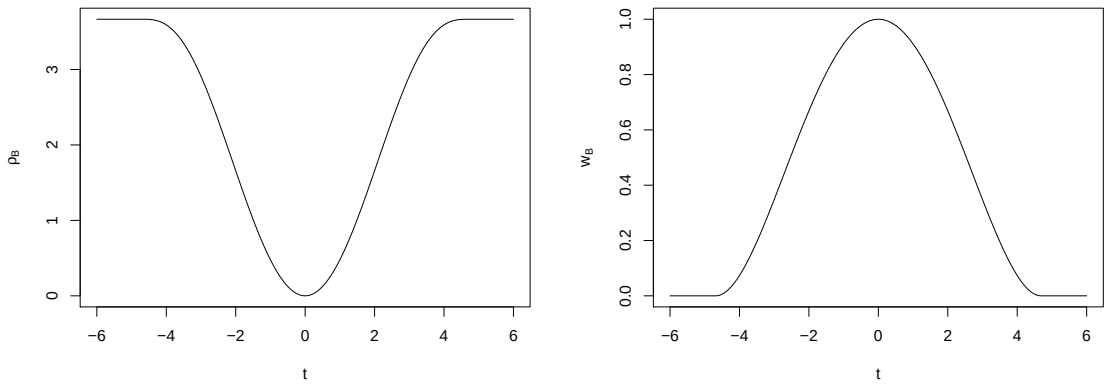

Figure 2.1 $\rho_{\mathbf{B}}$ and $w_{\mathbf{B}}$ functions for Tukey's biweight (bisquare) estimator.

2.2 High Dimensional Linear Regression

The methods are mentioned untill here are very useful for linear regression, but can not implemented when p exceeds n due to computational problems. Several altervatives have been proposed for this case. In this section, we give a short overview to some them.

2.2.1 PLS Estimator

A prominent biased estimator in case of $p > n$ is the Partial Least-Squares (PLS) estimator [29], which is based on modeling the predictor variables by means of a small set of latent variables. The latent variables are determined by maximizing the covariance between the response and a projection of the predictor variables, by employing appropriate orthogonality constraints.

More plainly, PLS regression aim to find a linear relation between the predictor variables and predictand vector, like (2.3), but rather than finding this relation directly, $\mathbf{X}$ and $\mathbf{y}$ are modeled by linear latent variables according to the regression models. Let us show predictor matrix $\mathbf{X}$ modeling by linear latent variables according to the regression model

as follows

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \boldsymbol{\varepsilon}_x \tag{2.12}$$

where $\boldsymbol{\varepsilon}_x$ is error matrix, $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_a]$ is the score matrix , which collects x-scores and can be considered as good summaries of the x-variables and $\mathbf{P}$ is the loading matrix with columns size of a, which can be estimated by CV. The relationship between x-scores and $\mathbf{y}$ becomes

$$\mathbf{y} = \mathbf{T}\mathbf{d} + \mathbf{h} \tag{2.13}$$

where $\mathbf{h}$ is the residuals and $\mathbf{d}$ corresponds to regression parameters. If for instance the linear relationship between $\mathbf{t}_1$ and $\mathbf{y}$ is strong (if the elements of $\mathbf{h}$ are small), then the x-score of the first PLS component is good for predicting $\mathbf{y}$. For more detail see [5].

So, PLS balances the maximal correlation criteria for OLS given in (2.4) with the requirement of explaining as much as variability in both x and y-space. Although PLS is useful for $p > n$, it is very sensitive to outlying observations.

2.2.2 Lasso Estimator

The lasso estimator is proposed by [7] adding an l_1 penalty to the objective function of LS as follows

$$\hat{\boldsymbol{\beta}}_{lasso} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \, |\boldsymbol{\beta}|_1 \right\} \tag{2.14}$$

for some $\lambda \geq 0$. Although this does no longer allow for a closed form solution for the estimated regression coefficients, the lasso estimator gets *sparse*, which means that some of the regression coefficients are shrunken to zero. This means that lasso acts like a variable selection method by returning a smaller subset of variables being relevant for the model. Therefore lasso can be regarded as variable selection method. Roughly speaking, adding an l_1 constrait for the coefficient estimates to the objective function yields sparse solutions.

This is appropriate in particular for high dimensional low sample size data sets ($n \ll p$), arising from applications in chemometrics, biometrics, econometrics, social sciences and many other fields, where the data include many uninformative variables which have no effect on the predictand or have very small contribution to the model.

2.2.3 Elastic Net Estimator

There is a limitation of the lasso estimator, since it is able to select only at most n variables when $n < p$. If n is very small, or if the number of informative variables (variables which are relevant for the model) is expected to be greater than n, the model performance can become poor. As a way out, the elastic net (*enet*) estimator has been introduced [8], which combines both l_1 and l_2 penalties:

$$\hat{\boldsymbol{\beta}}_{enet} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda P_\alpha(\boldsymbol{\beta}) \right\} \tag{2.15}$$

Here, $\mathbf{y} = (y_1, \ldots, y_n)^T$, the observations $\mathbf{x}_i^T$ form the rows of $\mathbf{X}$, and the penalty term P_α is defined as

$$P_\alpha(\boldsymbol{\beta}) = (1-\alpha)\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} \left[(1-\alpha)\frac{1}{2}\beta_j^2 + \alpha|\beta_j| \right]. \tag{2.16}$$

The entire strength of the penalty is controlled by the tuning parameter $\lambda \geq 0$. The other tuning parameter α is the mixing proportion of the ridge and lasso penalties and takes value in $[0, 1]$. The elastic net estimator is able to select variables like in lasso regression, and shrink the coefficients according to ridge. For an overview of sparse methods, see [9].

2.2.4 PRM Estimator

PLS is vey useful for $p > n$, not it loose much of its power in case of contaminated data set. Therefore, a robust version named Partial Robust M-estimator (PRM) has been proposed, which is based on M-estimation on latent variables, and also downweights leverage points [22]. In the PRM algorithm, two types of weights strategy are taken to obtain a total

robustness, namely weights for the residuals w_i^r as well as weights for the leverage points w_i^x, which are measuring the leverage of each score vector, are calculated seperately, then w_i can be combined to as follows

$$w_i = w_i^r w_i^x \quad \text{for} \quad i = 1, \ldots, n. \tag{2.17}$$

An iterative reweighted partial least squares algorithm is used to calculate PRM [22].

2.2.5 M-Liu Estimator

One proposal to robustify the Liu estimator is to employ the M-estimator, namely shrinking the M-estimator instead of the LS estimator, and the resulting robustified Liu estimator is defined as

$$\hat{\boldsymbol{\beta}}_{\mathrm{M-Liu}} = (\mathbf{X}'\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{X} + \lambda_{\mathrm{M}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\mathrm{M}}, \tag{2.18}$$

with the biasing parameter λ_{M}, see [3]. Since the M-estimator is not robust against leverage points, the M-Liu estimator has the same drawback.

2.2.6 LTS-Liu Estimator

Another proposal to robustify the Liu estimator is to use LTS estimator and the resulting estimator is called the LTS-Liu estimator. LTS-Liu estimator is defined as

$$\hat{\boldsymbol{\beta}}_{\mathrm{LTS-Liu}} = (\mathbf{X}'\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{X} + \lambda_{\mathrm{LTS}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\mathrm{LTS}}, \tag{2.19}$$

with the biasing parameter λ_{LTS} [4]. The LTS-estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LTS}}$ is robust against both y- and x-outliers. However, there is a question whether this robustness also implies robustness of the LTS-Liu estimator.

2.2.7 Sparse LTS Linear Regression

The sparse LTS regression estimator has been proposed for high dimensional problems [13]:

$$\hat{\boldsymbol{\beta}}_{\text{sparseLTS}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{h} r^2_{(i)}(\boldsymbol{\beta}) + h\lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{2.20}$$

This estimator adds an l_1 penalty to the objective function of the LTS estimator, and it can thus be seen as a robust counterpart of the lasso estimator. The idea of the calculation is inspried from the LTS estimator. Namely, the sparse LTS corresponds to finding the subset of $h \leq n$ observations whose lasso fit produces the smallest penalized residual sum of squares. This optimal subset is calculated by an analogue of the FAST-LTS algorithm [12]. The key feature of this algorithm is the "concentration step" or C-step, which is an efficient way to arrive at outlier-free data subsets where the lasso estimator can be applied. The FAST-LTS algorithm for LTS estimator used elemental subsets of size p, since any LS regression requires at least as many observations as the dimension p. But when the lasso estimator is used instead of LS, there is a situation that the data can include more variables than the observations, namely $p > n$, and this would make the algorithm not applicable. Fortunately the lasso is already well defined for samples of size 3, even for large values of p. Therefore, the elemental subsets of size 3 are only used to construct the initial subsets of size h for the C-step algorithms and then C-steps are continued on the subsets of size h till converge.

The sparse LTS estimator is robust to both vertical outliers and leverage points, and also a fast algorithm has been developed for its computation [30]. But the sparse LTS can suffer from the same problem as LTS, namely a low efficiency. To improve efficiency, a reweigthing step is carried out.

2.3 Logistic Regression

With a binary predictand vector $\mathbf{y}$ coded in the form $y_i \in \{0,1\}$, let us consider the logistic regression model

$$y_i = \pi_i + \varepsilon_i, \quad i = 1,\ldots,n, \tag{2.21}$$

where π_i denotes the conditional probability for observation i

$$\pi_i = Pr(y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\mathbf{x}_i^T\boldsymbol{\beta}}}{1+e^{\mathbf{x}_i^T\boldsymbol{\beta}}} \tag{2.22}$$

and ε_i is the error term assumed to have binomial distribution. Explicitly, the binary logistic regression can be regarded as a generalized linear model determined by the *logit* transformation

$$log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^T\boldsymbol{\beta}$$

as a link function [31]. Therefore, the transformation of the model ensures that probabilities lie between 0 and 1. The most popular way to estimate the model parameters is the maximum likelihood (ML) estimator which is based on maximizing the log-likelihood function or, equivalently, minimizing the negative log-likelihood function,

$$\hat{\boldsymbol{\beta}}_{\mathbf{ML}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} d(\mathbf{x}_i^T\boldsymbol{\beta}, y_i), \tag{2.23}$$

with the deviances

$$d(\mathbf{x}_i^T\boldsymbol{\beta}, y_i) = -y_i \log \pi_i - (1-y_i)\log(1-\pi_i) = -y_i\mathbf{x}_i^T\boldsymbol{\beta} + \log\left(1+e^{\mathbf{x}_i^T\boldsymbol{\beta}}\right). \tag{2.24}$$

The estimation of the model parameters with this method is not reliable when there is multicollinearity among the predictors and is not feasible when $p > n$ because of the need to invert near-singular and ill-conditioned information matrices.

2.3.1 Elastic Net Estimator

To deal with multicollinearity problem among the predictors and to provide a feasible solution in case of high dimensional data, Friedman et al. [16] suggested a new method which is based on minimization of a penalized negative log-likelihood function,

$$\hat{\boldsymbol{\beta}}_{enet} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} d(\mathbf{x}_i^T \boldsymbol{\beta}, y_i) + n\lambda P_\alpha(\boldsymbol{\beta}) \right\}. \tag{2.25}$$

Here, $P_\alpha(\boldsymbol{\beta})$ is the elastic net penalty as given in Equation (2.16), and thus this estimator extends (2.15) to the logistic regression setting. Using the elastic net penalty also solves the non-existence problem of the estimator in case of non-overlapping groups [19, 16, 20].

Nevertheless, the estimator (2.25) is not robust, and thus the results are badly affected in presence of outliers. In context of logistic regression, the definition of the vertical outliers needs to be re-defined because of the categorical predictand values. Therefore, misclassified observations are called vertical outliers.

2.3.2 BY Estimator

A highly robust estimator, the so called the BY estimator, is suggested by Bianco and Yohai [18] as follows

$$\hat{\boldsymbol{\beta}}_{\mathbf{BY}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \varphi(\mathbf{x}_i^T \boldsymbol{\beta}; y_i) \tag{2.26}$$

where φ denotes a positive and almost everywhere differentiable function. The BY estimator was well analysed by Croux and Haesbroeck [5]. For easier notation they define the univariate function $\phi_{\mathbf{BY}}(\mathbf{x}_i^T \boldsymbol{\beta})$, which corresponds to $y_i = 0$, instead of $\varphi(\mathbf{x}_i^T \boldsymbol{\beta}; y_i)$ since $\varphi(\mathbf{x}_i^T \boldsymbol{\beta}; 0) = \varphi(-\mathbf{x}_i^T \boldsymbol{\beta}; 1)$ for any observation i. With $s = \mathbf{x}_i^T \boldsymbol{\beta}$, the explicit formulation of $\phi_{\mathbf{BY}}(s)$ is as follows

$$\phi_{\mathbf{BY}}(s) = \rho\left(-\ln\left(1 - F(s)\right)\right) + G\left(F(s)\right) + G\left(1 - F(s)\right) - G(1) \tag{2.27}$$

where F presents the increasing cumulative distribution function defined by $F(s) = 1/(1+e^{-s})$ and

$$G(t) = \begin{cases} te^{-\sqrt{-\ln t}} + e^{\frac{1}{4}}\sqrt{\pi}\Phi\left(\sqrt{2}\left(\frac{1}{2}+\sqrt{-\ln t}\right)\right) - e^{-\frac{1}{4}}\sqrt{\pi} & if\ t \le e^{-d} \\ e^{-\sqrt{-d}}t - e^{\frac{1}{4}}\sqrt{\pi} + e^{-\frac{1}{4}}\sqrt{\pi}\Phi\left(\sqrt{2}\left(\frac{1}{2}+\sqrt{-d}\right)\right) & otherwise \end{cases} \tag{2.28}$$

where the constant d is the tuning parameter to compromise between robustness and efficiency and Φ is the normal cumulative distribution. The function $\phi_{\mathbf{BY}}(s)$ in equation (2.27) depends on the function ρ, therefore the choice of function ρ has a critical effect on BY estimator. Croux and Haesbroeck [17] suggested using the following ρ function.

$$\rho(t) = \begin{cases} te^{-\sqrt{d}} & if\ t \le d \\ -2e^{-\sqrt{t}}(1+\sqrt{t}) + e^{-\sqrt{d}}(2(1+\sqrt{d})+d) & otherwise \end{cases} \tag{2.29}$$

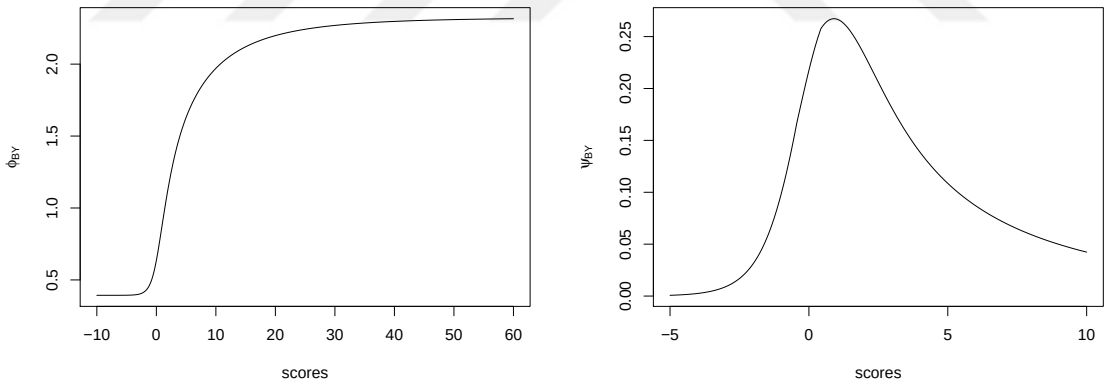

Figure 2.2 $\phi_{\mathbf{BY}}$ and $\psi_{\mathbf{BY}}$ functions for Bianco and Yohai estimator correspond to ρ function in Eq. (2.29).

The function $\phi_{BY}(s)$ yields large but bounded values for large positive scores (which correspond to misclassified observations). Therefore, the derivative of the function $\phi_{BY}(s)$ avoids to downweight misclassified observations too severely. Those effects of ϕ_{BY} and ψ_{BY} are displayed in Figure 2.2. For more information, see [17].

CHAPTER 3

ROBUST LINEAR REGRESSION

In this chapter two robust Liu-type estimators will be introduced: the first one is the MM-Liu estimator which inherits the property of MM estimator, so robust to both types of outliers and, the second one is the PRM-Liu estimator which is useful in case of high dimensional data.

3.1 MM-Liu Estimator

A fully robust version of a Liu estimator also needs to be robust for the choice of the biasing parameter. Consider the general form of a Liu-type estimator as given in Eq. (2.7) with an unbiased plug-in estimator $\hat{\boldsymbol{\beta}}$. Thus, $\mathrm{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$, with the true parameter vector $\boldsymbol{\beta}$. The biasing parameter λ can be determined by a mean-squared error (MSE) criterion [3]. Denote the bias of this Liu estimator as $bias(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}})$ and the covariance by $cov(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}})$. Then the MSE is defined as

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}}, \lambda) = bias(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}})' bias(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}}) + tr(cov(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}})) \tag{3.1}$$

where *tr* denotes the trace. Simple calculus shows that

$$bias(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}}) = \mathrm{E}(\hat{\boldsymbol{\beta}}_{\mathrm{Liu}} - \boldsymbol{\beta}) = (\mathbf{X}'\mathbf{X} + \mathbf{I})^{-1}(\lambda - 1)\boldsymbol{\beta} \tag{3.2}$$

[see 2].

The appropriate choice of λ is for the minimum value of the MSE in equation (3.1). It can

be seen immediately that leverage points affect this MSE criterion, and thus the choice of λ may be misleading. Similar arguments were given by [32] in the context of robust ridge regression, and a robustification of this criterion can be made by introducing weights for the observations. Fortunately, a robust plug-in estimator for (2.7) also provides weights w_i for each of the n observations, which can be arranged in the diagonal of the matrix $\mathbf{W} = \text{diag}(w_1, \ldots, w_n)$. The weights are based on the scaled residuals, $w_i = W(r_i/\hat{\sigma})$, using the weight function $W(r) = \rho'(r)/r$, where ρ' is the derivative of the ρ-function used in (2.9), see [10].

Accordingly, we propose the following robust Liu estimator,

$$\hat{\boldsymbol{\beta}}_{\text{MM-Liu}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{\text{MM}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\text{MM}} \tag{3.3}$$

which uses the MM-estimator $\hat{\boldsymbol{\beta}}_{\text{MM}}$ as plug-in estimator, as well as the weights $\mathbf{W}$ resulting from the MM-estimator [33]. Note that these weights are taken after MM-estimation and thus fixed for the robust Liu estimator.

3.1.1 Selection of the Tuning Parameter

For the optimal choice for the biasing parameter λ_{MM} we use two ways. One is to minimize the MSE given by following equation

$$\text{MSE}(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}, \lambda_{\text{MM}}) = bias(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}})' bias(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}) + tr(cov(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}})), \tag{3.4}$$

where

$$bias(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}) = \text{E}(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}} - \boldsymbol{\beta}) = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{I})^{-1}(\lambda_{\text{MM}} - 1)\boldsymbol{\beta}. \tag{3.5}$$

The use of weights robustifies the bias calculation and thus leads to a robust choice of the biasing parameter. The optimal biasing parameter is then calculated as follows [33]

$$\hat{\lambda}_{\text{MM}} = \underset{\lambda_{\text{MM}}}{\arg\min}\, \text{MSE}(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}, \lambda_{\text{MM}}). \tag{3.6}$$

Note that $tr(cov(\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}))$ needs to be estimated robustly. This is done by using the procedure proposed in [17].

There is also another aspect of using weights. Consider the singular value decomposition of the $n \times p$ matrix $\mathbf{X}$, with $n > p$, $\mathbf{X} = \mathbf{UDV}'$, with the singular values $d_1, \ldots, d_p$ in the diagonal of $\mathbf{D}$, the orthonormal columns $\mathbf{u}_1, \ldots, \mathbf{u}_p$ of $\mathbf{U}$, and the orthonormal columns $\mathbf{v}_1, \ldots, \mathbf{v}_p$ of $\mathbf{V}$. Using equation (2.7), the predicted values are

$$\hat{\mathbf{y}}_{\text{Liu}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{Liu}} = \mathbf{UD}(\mathbf{D}^2 + \mathbf{I})^{-1}(\mathbf{D}^2 + \lambda\mathbf{I})\mathbf{V}'\hat{\boldsymbol{\beta}} = \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j(d_j^2 + \lambda)}{d_j^2 + 1}\mathbf{v}_j'\hat{\boldsymbol{\beta}}. \tag{3.7}$$

On the other hand, the predicted values from the MM-Liu estimator (3.3) are

$$\hat{\mathbf{y}}_{\text{MM-Liu}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{MM-Liu}} = \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j(d_j(\mathbf{u}_j'\mathbf{W}\mathbf{u}_j)d_j + \lambda_{\text{MM}})}{d_j(\mathbf{u}_j'\mathbf{W}\mathbf{u}_j)d_j + 1}\mathbf{v}_j'\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}. \tag{3.8}$$

Outlying "scores" in $\mathbf{U}$ are downweighted by $\mathbf{u}_j'\mathbf{W}\mathbf{u}_j$ in equation (3.8), which is not the case in the unweighted version (3.7). This has an effect on the amount of shrinkage, and it can thus be seen as robustifying the shrinkage with respect to leverage points. In this case, we investigate the performance of the $\hat{\boldsymbol{\beta}}_{\text{MM-Liu}}$ by

$$\text{MSE}_(\hat{\mathbf{y}}_{\text{MM-Liu}}, \lambda_{\text{MM}}) = \frac{1}{n}\left[(\mathbf{y} - \hat{\mathbf{y}}_{\text{MM-Liu}})'(\mathbf{y} - \hat{\mathbf{y}}_{\text{MM-Liu}})\right]. \tag{3.9}$$

On the other hand, we also use cross-validation (CV) with $k = 5$. In more detail, for k-fold CV, the data are randomly split into k blocks of approximately equal size. Each block is left out once, the model is fitted to the "training set" contained in the $k-1$ blocks, and it is applied to the left-out block with the "test set". Therefore MSE calculated by means of 5-fold CV

$$\text{MSE}_{\text{CV}}(\hat{\mathbf{y}}^{\text{CV}}_{\text{MM-Liu}}, \lambda_{\text{MM}}) = \frac{1}{n}\left[(\mathbf{y} - \hat{\mathbf{y}}^{\text{CV}}_{\text{MM-Liu}})'(\mathbf{y} - \hat{\mathbf{y}}^{\text{CV}}_{\text{MM-Liu}})\right], \tag{3.10}$$

for comparing the performance of the estimator, since it gives more reliable results than the classical calculation method.

3.2 PRM-Liu Estimator

As mentioned before, the (classical) Liu estimator has been proposed for a situation with near multicollinearity. In case of collinearity, the unbiased LS-estimator could not be computed and used as a plug-in estimator. This problem typically occurs if the number of explanatory variables p is larger than the number of observations n. While the ridge estimator and various other biased estimators like Lasso [7] still work in this situation, one could also think about modifying the Liu estimator in order to cope with $p > n$ problems.

A prominent biased estimator in case of $p > n$ is the Partial Least-Squares (PLS) estimator [29], which is based on modeling the predictor variables by means of a small set of latent variables. The latent variables are determined by maximizing the covariance between the response and a projection of the predictor variables, by employing appropriate orthogonality constraints. Although PLS is useful for $p > n$, it is very sensitive to outlying observations. Therefore, a robust version named Partial Robust M-estimator (PRM) has been proposed, which is based on M-estimation on latent variables, and also downweights leverage points [22]. Thus, weights for the residuals w_i^r as well as weights for the leverage points w_i^x are computed, which can be combined to $w_i = w_i^r w_i^x$ for $i = 1, \ldots, n$. Using these weights in the diagonal of the matrix $\mathbf{W}$, the PRM-Liu estimator is defined as

$$\hat{\boldsymbol{\beta}}_{\text{PRM}-\text{Liu}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{\text{PRM}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\text{PRM}}, \tag{3.11}$$

where $\hat{\boldsymbol{\beta}}_{\text{PRM}}$ is the PRM estimator, and λ_{PRM} is the biasing parameter.

3.2.1 Selection of the Tuning Parameter

Selecting the biasing parameter λ_{PRM} according to a MSE error like in (3.4) is no longer possible in this case, since

$$bias(\hat{\boldsymbol{\beta}}_{\text{PRM}-\text{Liu}}) = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{\text{PRM}}\mathbf{I})\ \text{E}(\hat{\boldsymbol{\beta}}_{\text{PRM}}) - \boldsymbol{\beta}, \tag{3.12}$$

and $\mathrm{E}(\hat{\boldsymbol{\beta}}_{\mathrm{PRM}})$ cannot be derived analytically. Therefore, we will use CV to compute

$$\mathrm{MSE}_{\mathrm{CV}}(\hat{\mathbf{y}}^{\mathrm{CV}}_{\mathrm{PRM-Liu}}, \lambda_{\mathrm{PRM}}) = \frac{1}{n}\left[(\mathbf{y} - \hat{\mathbf{y}}^{\mathrm{CV}}_{\mathrm{PRM-Liu}})'(\mathbf{y} - \hat{\mathbf{y}}^{\mathrm{CV}}_{\mathrm{PRM-Liu}})\right], \tag{3.13}$$

where $\hat{\mathbf{y}}^{\mathrm{CV}}_{\mathrm{PRM-Liu}}$ are the predicted values using the PRM-Liu estimator for values of λ_{PRM} within a CV scheme. We use 5-fold CV for this purpose. Therefore the optimal biasing parameter is calculated by

$$\hat{\lambda}_{\mathrm{PRM}} = \underset{\lambda_{\mathrm{PRM}}}{\arg\min}\, \mathrm{MSE}(\hat{\mathbf{y}}^{\mathrm{CV}}_{\mathrm{PRM-Liu}}, \lambda_{\mathrm{PRM}}). \tag{3.14}$$

which corresponds to minimization of the Eq. (3.13).

CHAPTER 4

ROBUST AND SPARSE LINEAR REGRESSION

4.1 Robust and Sparse Linear Regression with Elastic Net Penalty

A robust and sparse elastic net estimator in linear regression can be defined with the objective function

$$Q(H,\boldsymbol{\beta}) = \sum_{i\in H}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + h\lambda P_\alpha(\boldsymbol{\beta}) \tag{4.1}$$

where $H \subseteq \{1,2,\ldots,n\}$ with $|H| = h$, $\lambda \in [0,\lambda_0]$, and P_α indicates the elastic net penalty with $\alpha \in [0,1]$ as in Equation (2.16). We call this estimator the *enet-LTS* estimator, since it uses a trimmed sum of squared residuals, like the sparse LTS estimator (2.20). The minimum of the objective function (4.1) determines the optimal subset of size h,

$$H_{opt} = \underset{H\subseteq 1,2,\ldots,n:|H|=h}{\arg\min} Q(H,\hat{\boldsymbol{\beta}}_H), \tag{4.2}$$

which is supposed to be outlier-free. The coefficient estimates $\hat{\boldsymbol{\beta}}_H$ depend on the subset H. For this subset H_{opt}, the enet-LTS estimator is given by

$$\hat{\boldsymbol{\beta}}_{enetLTS} = \arg\min Q(H_{opt},\boldsymbol{\beta}). \tag{4.3}$$

It is not trivial to identify this optimal subset, and practically one has to use an algorithm to approximate the solution. This algorithm uses C-steps: Suppose that the current h-subset in the kth iteration of the algorithm is denoted by H_k, and the resulting estimator by $\hat{\boldsymbol{\beta}}_{H_k}$.

Then the next subset H_{k+1} is formed by the indexes of those observations which correspond to the smallest h squared residuals

$$r_{k,i}^2 = (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{H_k})^2, \quad \text{for } i = 1, \ldots, n. \tag{4.4}$$

If $\hat{\boldsymbol{\beta}}_{H_{k+1}}$ denotes the estimator based on H_{k+1}, then by construction of the h-subsets it follows immediately:

$$Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_{k+1}}) \leq Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_k}) \leq Q(H_k, \hat{\boldsymbol{\beta}}_{H_k}) \tag{4.5}$$

This means that the C-steps decrease the objective function (4.1) successively, and lead to a local optimum after convergence. The global optimum is approximated by performing the C-steps with several initial subsets. However, in order to keep the runtime of the algorithm low, it is crucial that the initial subsets are chosen carefully. As motivated in [13], for a certain combination of the penalty parameters α and λ, elemental subsets are created consisting of the indexes of three randomly selected observations. Using only three observations increases the possibility of having no outliers in the elemental subsets. Let us denote these elemental subsets by

$$H_{el}^s = \{j_1^s, j_2^s, j_3^s\}, \tag{4.6}$$

where $s \in \{1, 2, \ldots, 500\}$. The resulting estimators based on the three observations are denoted by $\hat{\boldsymbol{\beta}}_{H_{el}^s}$. Now the squared residuals $(y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}_{H_{el}^s})^2$ can be computed for all observations $i = 1, \ldots, n$, and two C-steps are carried out, starting with the h-subset defined by the indexes of the smallest squared residuals. Then only those 10 h-subsets with the smallest values of the objective function (4.1) are kept as candidates. With these candidate subsets, the C-steps are performed until convergence (no further decrease), and the best subset is defined as that one with the smallest value of the objective function. This *best subset* also defines the estimator for this particular combination of α and λ.

Basically, one can apply this procedure now for a grid of values in the interval $\alpha \in [0, 1]$ and $\lambda \in [0, \lambda_0]$. Practically, this may still be quite time consuming, and therefore, for a

new parameter combination, the best subset of the neighboring grid value of α and/or λ, is taken, and the C-steps are started from this best subset until convergence. This technique, called *warm starts*, is repeated for each combination over the grid of α and λ values, and thus the start based on the elemental subsets is carried out only once.

Note that at the beginning of the algorithm for linear regression, the predictand is centered, and the predictor variables are centered robustly by the median and scaled by the MAD. Within the C-steps of the algorithm, we additionally mean-center the response variable and scale the predictors by their arithmetic means and standard deviations, calculated on each current subset, see also [13].

4.2 Selection of the Tuning Parameters

Section 4.1 outlined the algorithm to arrive at a best subset for robust elastic net linear regression, for each combination of the tuning parameters $\alpha \in [0,1]$ and $\lambda \in [0,\lambda_0]$. In this section we define the strategy to select the optimal combination α_{opt} and λ_{opt}, leading to the optimal subset. For this purpose we are using k-fold cross-validation (CV) on those best subsets of size h, with $k = 5$. In more detail, for k-fold CV, the data are randomly split into k blocks of approximately equal size. Each block is left out once, the model is fitted to the "training data" contained in the $k-1$ blocks, using a fixed parameter combination for α and λ, and it is applied to the left-out block with the "test data". In this way, h fitted values are obtained from k models, and they are compared to the corresponding original response by using the following evaluation criterion, which is the root mean squared prediction error (RMSPE)

$$\text{RMSPE}(\alpha,\lambda) = \sqrt{\frac{1}{h}\sum_{i=1}^{h} r_i^2(\hat{\boldsymbol{\beta}}_{\alpha,\lambda})} \tag{4.7}$$

where $r_i = y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{\alpha,\lambda}$ presents the test set residuals from the models estimated on the training sets with a specific α and λ (for simplicity we omitted here the index k denoting the models where the k-th block was left out and the corresponding test data from this block).

Note that the evaluation criterion given by (4.7) is robust against outliers, because it is based on the best subsets of size h, which are supposed to be outlier free.

In order to obtain more stable results, we repeat the k-fold CV five times and take the average of the corresponding evaluation measure. Finally, the optimal parameters α_{opt} and λ_{opt} are defined as that couple for which the evaluation criterion gives the minimal value. The corresponding best subset is determined as the optimal subset.

Note that the optimal couple α_{opt} and λ_{opt} is searched on a grid of values $\alpha \in [0,1]$ and $\lambda \in [0,\lambda_0]$. In our experiments we used 41 equally spaced values for α, and λ was varied in steps of size $0.025\lambda_0$. For determining λ_0 in the linear regression case we used the same approach as in Alfons et al. [13]. To clarify the idea behind, let us look at the following equation

$$\lambda_0 = \frac{2}{n} \max_{j \in \{1,\ldots,p\}} Cor(\mathbf{y}, \mathbf{x}_j), \tag{4.8}$$

where $Cor(\mathbf{y}, \mathbf{x}_j)$ presents the Pearson correlation between $\mathbf{y}$ and the jth predictor variable $\mathbf{x}_j$ of the design matrix $\mathbf{X}$. We took the robustified version of the Pearson correlation given in Eq. (5.5) as in Alfons et al. [13]. Here robustification is provided by calculating the Pearson correlation on bivariate winsorizated data [34].

4.3 Reweighting Step

The LTS estimator has a low efficiency, and thus it is common to use a reweighting step [11]. This idea is also used for the estimators introduced here. Generally, in a reweighting step the outliers according to the current model are identified and downweighted. For the linear regression model we will use the same reweighting scheme as proposed in Alfons et al. [13], which is based on standardized residuals. Explicitly, under the normal error model, observations with standardized residuals larger than a certain quantile of the standard normal distribution may be proclaimed as outliers. First step will be calculating the center

of the residuals. A common estimate for the center of the residuals is

$$\mu_{raw} = \frac{1}{h} \sum_{i \in H_{opt}} r_i, \quad h \in H_{opt} \tag{4.9}$$

where $r_i = y_i - \mathbf{x}_i \boldsymbol{\beta}_{enetLTS}$ and H_{opt} stands for the optimal subset given in Eq. 4.2. Then the residual scale estimate of the raw enet-LTS estimator is given by

$$\hat{\sigma}_{raw} = k_\alpha \sqrt{\frac{1}{h} \sum_{i=1}^{h} (r_c^2)_{1:n}}, \tag{4.10}$$

where $r_c^2 = ((r_1 - \hat{\mu}_{raw})^2, \ldots, (r_n - \hat{\mu}_{raw})^2)^T$ and k_α is a factor to garantee that raw $\hat{\sigma}$ is a consistent estimate of the standard deviation at the normal model and given with following equation

$$k_\alpha = \left(\frac{1}{\alpha} \int_{-\Phi^{-1}(\alpha+1)/2}^{\Phi^{-1}(\alpha+1)/2} u^2 d\Phi(u) \right)^{-1/2}. \tag{4.11}$$

For simplicity, we indicate the standardized residuals from the linear regression case by r_i^s. Then the weights are defined by

$$w_i = \begin{cases} 1, & \text{if } |r_i^s| \leq \Phi^{-1}(1-\delta) \\ 0, & \text{if } |r_i^s| > \Phi^{-1}(1-\delta) \end{cases} \quad i = 1, 2, \ldots, n, \tag{4.12}$$

where $\delta = 0.0125$, such that 2.5% of the observations are flagged as outliers in the normal model. The reweighted enet-LTS estimator is defined as

$$\hat{\boldsymbol{\beta}}_{reweighted} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} w_i f(\mathbf{x}_i; y_i) + \lambda_{upd} n_w P_{\alpha_{opt}}(\boldsymbol{\beta}) \right\}, \tag{4.13}$$

where w_i, $i = 1, \ldots, n$ stands for the vector of binary weights, $n_w = \sum_{i=1}^{n} w_i$, and f corresponds to squared residuals for linear regression. Since $h \leq n_w$, and because the optimal parameters α_{opt} and λ_{opt} have been derived with h observations, the penalty can act (slightly) differently in (4.13) than for the raw estimator. For this reason, the parameter λ_{opt} has to be updated, while the α_{opt} regulating the tradeoff between the l_1 and

l_2 penalty is kept the same. The updated parameter λ_{upd} is determined by 5-fold CV, with the simplification that α_{opt} is already fixed.

CHAPTER 5

ROBUST AND SPARSE LOGISTIC REGRESSION

5.1 Robust and Sparse Logistic Regression with Elastic Net Penalty

Based on the definition (2.25) of the elastic net logistic regression estimator, it is straightforward to define the objective function of its robust counterpart based on trimming,

$$Q(H,\boldsymbol{\beta}) = \sum_{i\in H} d(\mathbf{x}_i^T\boldsymbol{\beta}, y_i) + h\lambda P_\alpha(\boldsymbol{\beta}), \tag{5.1}$$

where again $H \subseteq \{1,2,\ldots,n\}$ with $|H| = h$, and P_α is the elastic net penalty as defined in Equation (2.16). As outlined in Section 4.1 for linear regression case, the task is to find the optimal subset which minimizes the objective function and defines the robust sparse elastic net estimator for logistic regression. It turns out that the algorithm explained previously in the linear regression setting can be successfully used to find the approximative solution. In the following we will explain the modifications that need to be carried out.

C-steps: In the linear regression case, the C-steps were based on the squared residuals (4.4). Now the h-subsets are determined according to the indexes of those observations with the smallest values of the deviances $d(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{H_k}, y_i)$. However, here it needs to be made sure that the original group sizes are in the same proportion. Denote n_0 and n_1 the number of observations in both groups, with $n_0 + n_1 = n$. Then $h_0 = \lfloor (n_0+1)h/n \rfloor$ and $h_1 = h - h_0$ define the group sizes in each h-subset. A new h-subset is created with the h_0 indexes of the smallest deviances $d(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{H_k}, y_i = 0)$ and with the h_1 indexes of the smallest deviances $d(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{H_k}, y_i = 1)$.

Elemental subsets: In the linear regression case, the elemental subsets consisted of the

indexes of three randomly selected observations, see (4.6). Now four observations are randomly selected to form the elemental subsets, two from each group. This allows to compute the estimator, and the two C-steps are based on the h smallest values of the deviances. As before, this is carried out for 500 elemental subsets, and only the "best" 10 h-subsets are kept. Here, "best" refers to an evaluation that is borrowed from a robustified deviance measure proposed in Croux and Haesbroeck [35] in the context of robust logistic regression (but not in high dimension). These authors replace the deviance function (2.24) used in (2.23) by a function φ_{BY} to define the so-called Bianco Yohai (BY) estimator

$$\hat{\boldsymbol{\beta}}_{BY} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \varphi(\mathbf{x}_i^T \boldsymbol{\beta}; y_i), \tag{5.2}$$

a highly robust logistic regression estimator, see also [18]. The form of the function φ_{BY} is shown in Figure 5.1, see [17] for details.

We use this function as follows: Positive scores $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}$ of group 1, i.e. $y_i = 1$, refer to correct classification and receive the highest values for φ_{BY}, while negative scores refer to misclassification, with small or zero φ_{BY} values. For the scores of group 0 we have the reverse behavior, see Figure 5.1. When evaluating an h-subset, the sum over the h values of $\varphi_{BY}(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_H)$ for $i \in H$ is computed, and this sum should be as large as possible. This means that we aim at identifying an h-subset where the groups are separated as much as possible. Points on the wrong side have almost no contribution, but also the contribution of outliers on the correct side is bounded. In this way, outliers will not dominate the sum.

With the best 10 h-subsets we continue the C-steps until convergence. Finally, the subset with the largest sum $\varphi_{BY}(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_H)$ over all $i \in H$ forms the best index set.

The selection of the optimal parameters α_{opt} and λ_{opt} is discussed in Section 5.2. The subset corresponding to these optimal tuning parameters is defined as the optimal subset of size h. The enet-LTS logistic regression estimator is then calculated on the optimal subset with α_{opt} and λ_{opt}.

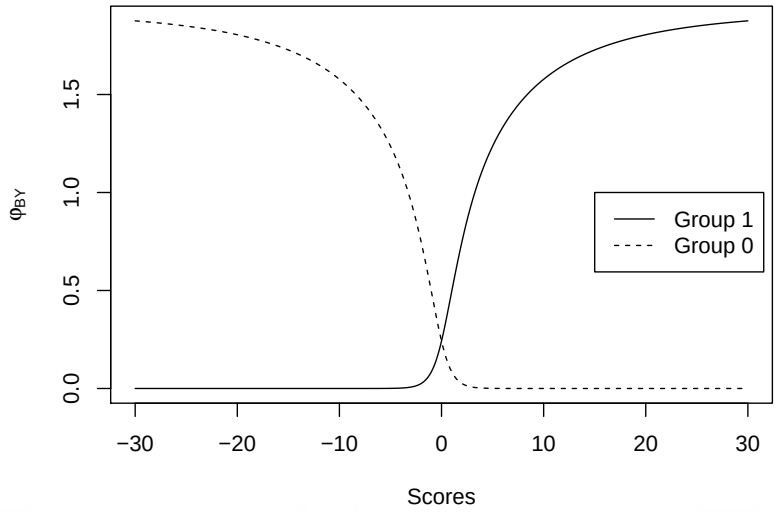

Figure 5.1 Function φ_{BY} used for evaluating an h-subset, based on the scores $\mathbf{x}_i^T\hat{\boldsymbol{\beta}}$ for the two groups.

The same procedure as linear regression is applied for logistic regression, except for centering the predictand. In the end, the coefficients are back-transformed to the original scale.

5.2 Selection of the Tuning Parameters

Section 5.1 outlined the algorithm to arrive at a best subset for robust elastic net logistic regression, for each combination of the tuning parameters $\alpha \in [0,1]$ and $\lambda \in [0,\lambda_0]$. In this section we define the strategy to select the optimal combination α_{opt} and λ_{opt}, leading to the optimal subset. For this purpose we are using k-fold cross-validation (CV) on those best subsets of size h, with $k = 5$. In more detail, for k-fold CV, the data are randomly split into k blocks of approximately equal size. In case of logistic regression, each block needs to consist of observations from both classes with approximately the same class proportions as in the complete data set. Each block is left out once, the model is fitted to the "training data" contained in the $k-1$ blocks, using a fixed parameter combination for α and λ, and it is applied to the left-out block with the "test data". In this way, h fitted values are obtained from k models, and they are compared to the corresponding original response by using the following evaluation criteria:

For logistic regression we use the mean of the negative log-likelihoods or deviances (MNLL)

$$\text{MNLL}(\alpha,\lambda) = \frac{1}{h}\sum_{i=1}^{h} d_i(\hat{\boldsymbol{\beta}}_{\alpha,\lambda}), \tag{5.3}$$

where $d_i = d(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{\alpha,\lambda}, y_i)$ presents the test set deviances from the models estimated on the training sets with a specific α and λ.

Note that the evaluation criterion given by (5.3) is robust against outliers, because it is based on the best subsets of size h, therefore it is supposed to be outlier free.

In order to obtain more stable results, we repeat the k-fold CV five times and take the average of the corresponding evaluation measure. Finally, the optimal parameters α_{opt} and λ_{opt} are defined as that couple for which the evaluation criterion gives the minimal value. The corresponding best subset is determined as the optimal subset.

Note that the optimal couple α_{opt} and λ_{opt} is searched on a grid of values $\alpha \in [0,1]$ and $\lambda \in [0,\lambda_0]$. In our experiments we used 41 equally spaced values for α, and λ was varied in steps of size $0.025\lambda_0$. For determining λ_0 in logistic regression we replaced the Pearson correlation, which is used in case of linear regression, by a robustified point-biserial correlation. Explicitly, we denote the group sizes of the two groups by n_0 and n_1, and by m_j^0 and m_j^1 the medians of the jth predictor variable for the data from the two groups, respectively. Then the robustified point-biserial correlation between $\mathbf{y}$ and $\mathbf{x}_j$ is defined as

$$r_{pb}(\mathbf{y},\mathbf{x}_j) = \frac{m_j^1 - m_j^0}{\text{MAD}(\mathbf{x}_j)} \cdot \sqrt{\frac{n_0 n_1}{n(n-1)}}, \tag{5.4}$$

where $\text{MAD}(\mathbf{x}_j)$ is the MAD of $\mathbf{x}_j$, and $n = n_0 + n_1$. Finally, the border of interval for λ will be determined using following equation

$$\lambda_0 = \frac{2}{n}\max_{j\in\{1,\ldots,p\}} r_{pb}(\mathbf{y},\mathbf{x}_j), \tag{5.5}$$

where the correlation is calculated on bivariate winsorized data to obtain robust version.

5.3 Reweighting Step

The LTS estimator has a low efficiency, and thus it is common to use a reweighting step [11]. We also take into consideration this idea for the estimator introduced here. Generally, in a reweighting step the outliers are identified and downweighted. In case of logistic regression we compute the Pearson residuals which are approximately standard normally distributed and given by

$$r_i^s = \frac{y_i - \pi_i}{\pi_i(1-\pi_i)}, \tag{5.6}$$

with π_i the conditional probabilities from (2.22).

Then the weights are defined by

$$w_i = \begin{cases} 1, & \text{if } |r_i^s| \leq \Phi^{-1}(1-\delta) \\ 0, & \text{if } |r_i^s| > \Phi^{-1}(1-\delta) \end{cases} \qquad i = 1,2,\ldots,n, \tag{5.7}$$

where $\delta = 0.0125$, such that 2.5% of the observations are flagged as outliers in the normal model. The reweighted enet-LTS estimator is defined as

$$\hat{\boldsymbol{\beta}}_{reweighted} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} w_i f(\mathbf{x}_i; y_i) + \lambda_{upd} n_w P_{\alpha_{opt}}(\boldsymbol{\beta}) \right\}, \tag{5.8}$$

where w_i, $i = 1,\ldots,n$ stands for the vector of binary weights (according to the current model), $n_w = \sum_{i=1}^{n} w_i$, and f corresponds to squared residuals for linear regression or to the deviances in case of logistic regression. Since $h \leq n_w$, and because the optimal parameters α_{opt} and λ_{opt} have been derived with h observations, the penalty can act (slightly) differently in (5.8) than for the raw estimator. For this reason, the parameter λ_{opt} has to be updated, while the α_{opt} regulating the tradeoff between the l_1 and l_2 penalty is kept the same. The updated parameter λ_{upd} is determined by 5-fold CV, with the simplification that α_{opt} is already fixed.

CHAPTER 6

SIMULATIONS

In this chapter, some simulation studies are conducted to demonstrate the goodness of the proposed estimators. There are three main parts for each proposed estimators. First part gives the details of the generated data sets. Second part introduces which performance measures are used to compare the proposed estimators. Finaly, the results are shown in last part.

6.1 Simulation Studies for Robust Linear Regression

6.1.1 Sampling Schemes for Robust Regression

Let us describe sampling schemes by means of generating a "low dimensional" data set with $n = 50$ and $p = 10$ and a "high dimensional" data set with $n = 30$ and $p = 40$. The simulated data sets are generated in analogy to [32] for robust ridge regression, which consists of the following steps: The explanatory variables $(\mathbf{x}_1, \ldots, \mathbf{x}_p)$ are generated from a normal distribution $\mathcal{N}_p(\mathbf{0}, \mathbf{V})$, where $\mathbf{V} = [v_{jk}]$ with $v_{jj} = 1$ and $v_{jk} = \rho$ for $j \neq k$, $j,k = 1, \ldots, p$. Random errors e_i are generated from standard normal distribution $\mathcal{N}_p(0,1)$. The observations of the response variable are then determined by

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta}_{\text{true}} + e_i, \tag{6.1}$$

where $\boldsymbol{\beta}_{\text{true}} = \sqrt{pR}\mathbf{b}$. Note that $\boldsymbol{\beta}_{\text{true}}$ is determined randomly because $\mathbf{b}$ has a uniform distribution with $\mathbf{b}^T\mathbf{b} = 1$ and where the "Signal to Noise Ratio" (SNR) is given by

$R = \frac{\|\boldsymbol{\beta}_{\text{true}}\|^2}{p\text{Var}(e)}$. The first m observations $(\mathbf{x}_i, y_i)$ are changed to leverage points, where $m = [n\varepsilon]$ and $\varepsilon \in (0,1)$, and $[\cdot]$ denotes the integer part. Their magnitude is controlled by the parameters for leverage (k_{lev}) and slope (k_{slo}) through $\mathbf{x}_i = \mathbf{x}_0$ and $y_i = \mathbf{x}_0^T \boldsymbol{\beta}_{\text{true}}(1 + k_{slo})$, where $\mathbf{x}_0 = k_{lev}\mathbf{a}/\sqrt{\mathbf{a}^T\mathbf{V}^{-1}\mathbf{a}}$ is selected randomly with $\mathbf{a}^T\mathbf{1}_p = 0$. Therefore, $\mathbf{x}_0$ creates leverage points that are most influential to the estimator. The values for k_{slo} are modified in a grid in order to produce the largest MSE for each estimator.

Since the estimators PLS [29] and PRM [22] have good properties when $p > n$, we also take into account these estimators for comparisons. Moreover, we also compare with the non-robust PLS-Liu estimator, which is introduced in analogy to the PRM-Liu estimator as

$$\hat{\boldsymbol{\beta}}_{\text{PLS}-\text{Liu}} = (\mathbf{X}'\mathbf{X} + \mathbf{I})^{-1}(\mathbf{X}'\mathbf{X} + \lambda_{\text{PLS}}\mathbf{I})\hat{\boldsymbol{\beta}}_{\text{PLS}}, \tag{6.2}$$

where λ_{PLS} presents the biasing parameter, and $\hat{\boldsymbol{\beta}}_{\text{PLS}}$ is the PLS estimator.

We decide to take $\rho = 0.9$, $\varepsilon \in \{0, 0.1, 0.2\}$, and $k_{lev} = 10$, since these already allow to get a general picture of the performance of the different estimators. We choose SNR $\in \{0.1, 1, 10\}$ for low dimensional data and SNR $\in \{0.1, 0.5, 1\}$ for high dimensional data. Note that three contamination levels are provided for a wide perspective to highly contaminated data from uncontaminated data. The first case corresponds to the uncontaminated data, the second case includes 10% outliers and the third case has 20% outliers; both scenarios might be quite realistic in practice. The range of values for k_{slo} is limited to 30, which is sufficient to deliver the maximum of the MSE for the robust estimators. Note, however, that the MSE of the classical estimators could be increased artificially just by increasing k_{slo}, and thus the MSE will not be reported for classical estimators if $\varepsilon > 0$.

The simulated explanatory variables are first mean-centered and normalized to unit scale. For the classical estimators, mean and standard deviation are used, while for the robust estimators we use median and median absolute deviation (MAD). After estimation, the regression parameters are back-transformed to the original scale.

6.1.2 Performance Measures

The performance of the estimator $\hat{\boldsymbol{\beta}} = \left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}_1^T\right)^T$ is evaluated by

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}, \hat{\lambda}) = E(y_0 - \hat{\beta}_0 - \mathbf{x}_0^T \hat{\boldsymbol{\beta}}_1)^2 = 1 + \Delta, \tag{6.3}$$

where

$$\Delta = \hat{\beta}_0^2 + (\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{\mathrm{true}})^T \mathbf{V} (\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{\mathrm{true}})$$

and $\hat{\lambda}$ is the optimized Liu parameter, see equations (3.6) and (3.14). In addition, the prediction performance of the model using $\hat{\boldsymbol{\beta}}$ with the optimized parameter $\hat{\lambda}$ to predict the response $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ is evaluated through 5-fold cross-validation by

$$\mathrm{MSE}_{\mathrm{CV}}(\hat{\mathbf{y}}, \hat{\lambda}) = \frac{1}{n} \left[(\mathbf{y} - \hat{\mathbf{y}}^{\mathrm{CV}})^T (\mathbf{y} - \hat{\mathbf{y}}^{\mathrm{CV}}) \right]. \tag{6.4}$$

Each setting in the simulation study is repeated 200 times. The resulting MSE values are summarized by the 10% upper trimmed mean, because typically robust estimators are characterized by heavy-tailed distributions.

6.1.3 Results for Robust Linear Regression

All results given below correspond to this trimmed MSE for both contaminated and uncontaminated data sets for reasons of comparability.

Before presenting the simulation results, we first provide more insight into the design of the simulation. Figure 6.1 shows the resulting MSEs according to equations (6.3) (left picture) and (6.4) (right picture) for one simulated data set with $p = 10$, $n = 50$, SNR$= 1$, and $\varepsilon = 0.1$. Here the values for k_{slo} are varied between 0 and 30. The LS-Liu estimator increases quickly with a larger value of k_{slo}, and it would become unbounded for k_{slo} tending to infinity. The robust estimators are bounded, and for values of k_{slo} larger than 5 the values do almost not change any more. In the simulation tables below we will report only the maximum MSE values that are attained when varying the parameter k_{slo}. For the

non-robust LS-Liu estimator we will not report results in case of contamination, since the maximum MSE is infinity.

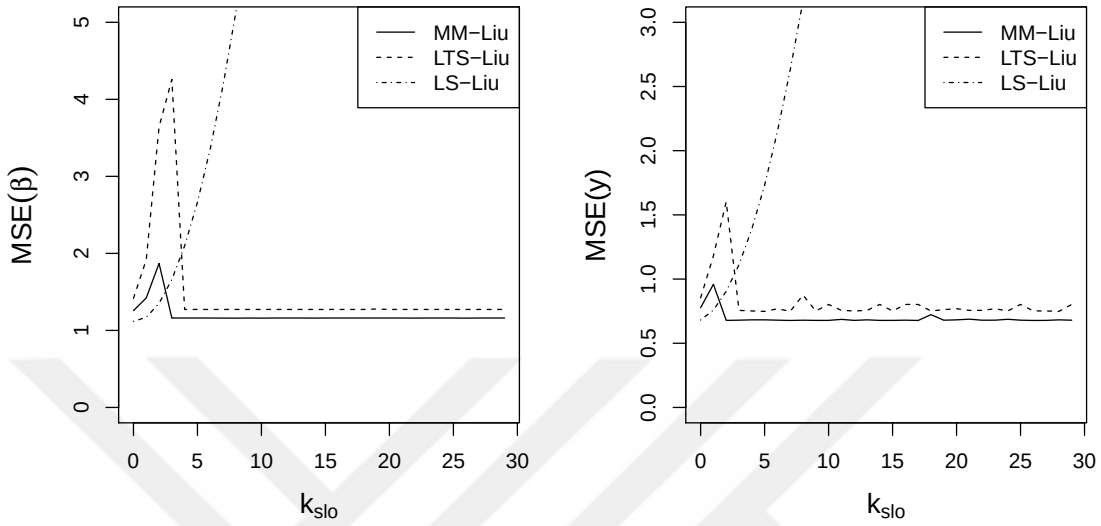

Figure 6.1 Simulated MSEs for $n = 50$, $p = 10$, SNR= 1, and $\varepsilon = 0.1$, for the MSE according to (6.3) (left) and (6.4) (right), for the MM-Liu, LTS-Liu and LS-Liu estimators, as a function of contamination slope.

Table 6.1 lists the results of the maximum trimmed MSEs calculated by (6.3), denoted as "(MSE)", and (6.4), denoted as "(CV)". We used the parameters $p = 10$, $n = 50$, and different values for SNR. This table represents the uncontaminated case ($\varepsilon = 0$). The results for the contaminated case for $\varepsilon = 0.1$ and $\varepsilon = 0.2$ are shown in Table 6.2.

One can see from Table 6.1 that the LS-Liu estimator increases only slightly with increasing signal-to-noise ratio, whereas this increase is larger for the robust estimators. The MM-Liu estimator outperforms the LTS-Liu estimator with respect to both MSE measurements.

Table 6.1 Maximum trimmed MSEs of estimates for $p = 10$, $n = 50$, and ε =0.

	SNR		
	10	1	0.1
MM-Liu (MSE)	2.21	1.31	1.20
LTS-Liu (MSE)	2.60	1.53	1.37
LS-Liu (MSE)	1.24	1.18	1.17
MM-Liu (CV)	1.65	0.90	0.79
LTS-Liu (CV)	1.75	0.99	0.87
LS-Liu (CV)	0.76	0.74	0.75

Table 6.2 shows the results in presence of contamination. Generally, the MSEs clearly increase with higher SNR and higher contamination. There is, however, a clear advantage of the MM-Liu estimator over the LTS-Liu estimator. In particular the MSEs of $\hat{\boldsymbol{\beta}}_{\text{LTS}}$ increase enormously compared to those of $\hat{\boldsymbol{\beta}}_{\text{MM}}$. This behavior was already visible in Figure 6.1. The reason for the difference is not only due to the higher efficiency of the MM-estimator compared to the LTS-estimator, but also due to the use of weights within the definition of the MM-Liu estimator.

Table 6.2 Maximum trimmed MSEs of estimates for $p = 10$, $n = 50$ under contamination.

SNR:	10		1		0.1	
ε :	0.1	0.2	0.1	0.2	0.1	0.2
MM-Liu (MSE)	8.56	29.31	2.65	8.34	1.79	3.65
LTS-Liu (MSE)	20.39	142.97	5.52	34.47	3.46	13.30
MM-Liu (CV)	5.96	19.02	1.85	5.43	1.23	2.38
LTS-Liu (CV)	7.32	24.92	2.10	6.14	1.45	3.21

In a further simulation study we increase the ratio p/n by taking $p = 10$ and $n = 25$. The results are reported in Table 6.3 for the uncontaminated case, and in Table 6.4 for contamination with $\varepsilon = 0.1$. We come to quite similar conclusions as before: The robust estimators lead to a loss in efficiency compared to the LS counterpart, in particular for high SNR. In case of contamination, the MM-Liu estimator again outperforms the LTS-Liu estimator, and the difference gets more pronounced for higher SNR values.

Table 6.3 Maximum trimmed MSEs of estimates for $p = 10$, $n = 25$, and ε =0.

	SNR		
	10	1	0.1
MM-Liu (MSE)	3.95	1.81	1.54
LTS-Liu (MSE)	4.83	1.92	1.62
LS-Liu (MSE)	1.60	1.37	1.34
MM-Liu (CV)	2.06	0.80	0.66
LTS-Liu (CV)	2.46	1.00	0.83
LS-Liu (CV)	0.58	0.56	0.56

Table 6.5 shows the results for the uncontaminated case, whereas the results for the contaminated cases are in Table 6.6. Here the MSEs refer to values calculated according to (6.4), and again we use upper 10% trimming of the 200 simulation replications. Note that

Table 6.4 Maximum trimmed MSEs of estimates for $p = 10$, $n = 25$, and ε =0.1.

	SNR		
	10	1	0.1
MM-Liu (MSE)	14.69	4.84	3.32
LTS-Liu (MSE)	28.41	7.89	4.49
MM-Liu (CV)	5.49	1.72	1.19
LTS-Liu (CV)	8.01	2.33	1.53

here also the optimal number of PLS (PRM) components needs to be determined, which is done by 5-fold CV. The results from both tables reveal that the plug-in estimators PLS or PRM can not be improved within the Liu estimation. There are only slight differences visible, which are essentially due to different optimal numbers of components. Since this optimal number of components is determined by the smallest MSE using CV for each simulated data set, it can be different for the PLS (PRM) estimator and the PLS-Liu (PRM-Liu) estimator.

Table 6.5 Maximum trimmed MSEs of estimates for $p = 40$, $n = 30$, and $\varepsilon = 0$.

	SNR		
	1	0.5	0.1
PLS-Liu (CV)	0.6	0.7	0.7
PLS (CV)	0.7	0.7	0.7
PRM-Liu (CV)	2.3	1.4	1.0
PRM (CV)	2.2	1.4	1.0

Table 6.6 Maximum trimmed MSEs of estimates for $p = 40$, $n = 30$, under contamination.

SNR:	1		0.5		0.1	
ε :	0.1	0.2	0.1	0.2	0.1	0.2
PRM-Liu (CV)	8.5	20.6	5.2	9.7	1.8	3.4
PRM (CV)	8.8	20.6	5.5	9.5	1.8	3.4

6.2 Simulation Studies for Robust and Sparse Linear Regression

6.2.1 Sampling Schemes for Robust and Sparse Linear Regression

Let us consider two different scenarios by means of generating a "low dimensional" data set with $n = 150$ and $p = 60$ and a "high dimensional" data set with $n = 50$ and $p = 100$. We generate a data matrix where the variables are forming correlated blocks, $\mathbf{X} = (\mathbf{X}_{a_1}, \mathbf{X}_{a_2}, \mathbf{X}_b)$, where $\mathbf{X}_{a_1}$, $\mathbf{X}_{a_2}$ and $\mathbf{X}_b$ have the dimensions $n \times p_{a_1}$,$n \times p_{a_2}$ and $n \times p_b$, with $p = p_{a_1} + p_{a_2} + p_b$. Such a block structure can be assumed in many applications, and it mimics different underlying hidden processes. The observations of the blocks are generated independently from each other, from a multivariate normal distribution $\mathcal{N}_{p_{a_1}}(\mathbf{0}, \mathbf{\Sigma}_{a_1})$ with $\mathbf{\Sigma}_{a_1} = \rho_{a_1}^{|j-k|}$, $1 \leq j, k \leq p_{a_1}$, $\mathcal{N}_{p_{a_2}}(\mathbf{0}, \mathbf{\Sigma}_{a_2})$ with $\mathbf{\Sigma}_{a_2} = \rho_{a_2}^{|j-k|}$, $1 \leq j, k \leq p_{a_2}$, and $\mathcal{N}_{p_b}(\mathbf{0}, \mathbf{\Sigma}_b)$ with $\mathbf{\Sigma}_b = \rho_b^{|j-k|}$, $1 \leq j, k \leq p_b$, respectively. While the first two blocks belong to the informative variables with sizes of $p_{a_1} = 0.05p$ and $p_{a_2} = 0.05p$, the third block represents uninformative variables with $p_b = 0.9p$. Furthermore, we take $\rho_{a_1} = \rho_{a_2} = 0.9$ to allow for a high correlation among the informative variables, and $\rho_b = 0.2$ to have low correlation among the uninformative variables.

To create sparsity, the true parameter vector $\boldsymbol{\beta}$ consists of zeros for the last 90% of the entries referring to the uninformative variables, while the first 10% of the entries are assigned to one. The response variable is calculated by

$$y_i = 1 + \mathbf{x}_i^T \boldsymbol{\beta} + e_i, \tag{6.5}$$

where the error term e_i is distributed according to a standard normal distribution $\mathcal{N}(0,1)$, for $i = 1, \ldots, n$.

This is the design for the simulations with clean data. For the simulation scenarios with outliers we replace the first 10% of the observations of the block of informative variables by values coming from independent normal distributions $\mathcal{N}(20,1)$ for each variable. Further, the error terms for these 10% outliers are replaced by values from $\mathcal{N}(20\hat{\sigma}_y, 1)$ instead of $\mathcal{N}(0,1)$, where $\hat{\sigma}_y$ represents the estimated standard deviation of the clean predictand

vector. In this way, the contaminated data consist of both vertical outliers and leverage points.

6.2.2 Performance Measures

For the evaluation of the different estimators, training and test data sets are generated according to the explained sampling schemes. The models are fit to the training data and evaluated on the test data. The test data are always generated without outliers.

As performance measures we use the root mean squared prediction error (RMSPE) for linear regression,

$$\text{RMSPE}(\hat{\boldsymbol{\beta}}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}\right)^2}. \tag{6.6}$$

where y_i and $\mathbf{x}_i$, $i = 1,\ldots,n$, indicate the observations of the test data set, $\hat{\boldsymbol{\beta}}$ denotes the coefficient vector and $\hat{\beta}_0$ stands for the estimated intercept term obtained from the training data set. Further, we consider the precision of the coefficient estimate as a quality criterion, defined by

$$\text{PRECISION}(\hat{\boldsymbol{\beta}}) = \sqrt{\sum_{i=0}^{p}\left(\beta_i - \hat{\beta}_i\right)^2}, \tag{6.7}$$

In order to compare the sparsity of the coefficient estimators, we evaluate the False Positive Rate (FPR) and the False Negative Rate (FNR), defined as

$$\text{FPR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j = 0,\ldots,p : \hat{\beta}_j \neq 0 \wedge \beta_j = 0\}|}{|\{j = 0,\ldots,p : \beta_j = 0\}|}, \tag{6.8}$$

$$\text{FNR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j = 0,\ldots,p : \hat{\beta}_j = 0 \wedge \beta_j \neq 0\}|}{|\{j = 0,\ldots,p : \beta_j \neq 0\}|}. \tag{6.9}$$

The FPR is the proportion of non-informative variables that are incorrectly included in the model. On the other hand, the FNR is the proportion of informative variables that are incorrectly excluded from the model. A high FNR usually has a bad effect on the

prediction performance since it inflates the variance of the estimator.

These evaluation measures are calculated for the generated data in each of 100 simulation replications separately, and then summarized by boxplots. The smaller the value for these criteria, the better the performance of the method.

6.2.3 Results for Robust and Sparse Linear Regression

The outcome of the simulations for linear regression is summarized in Figures 6.2–6.5. The left plots in these figures are for the simulations with low dimensional data, and the right plots for the high dimensional configuration. Figure 6.2 compares the RMSPE. All methods yield similar results in the low dimensional non-contaminated case, while in the high dimensional clean data case the elastic net method is clearly better. However, in the contaminated case, elastic net leads to poor performance, which is also the case for sparse LTS. Enet-LTS performs even slightly better with contaminated data, and there is also a slight improvement visible in the reweighted version of this estimator. The PRECISION in Figure 6.3 shows essentially the same behavior. The FPR in Figure 6.4, reflecting the proportion of incorrectly added noise variables to the models, shows a very low rate for sparse LTS. Here, the elastic net even improves in the contaminated setting, and the same is true for enet-LTS. A quite different picture is shown in Figure 6.5 with the FNR. Sparse LTS and elastic net miss a high proportion of informative variables in the contaminated data scenario, which is the reason for their poor overall performance. Note that the outliers are placed in the informative variables, which seems to be particularly difficult for sparse LTS.

In Table 6.7 and 6.8, the averaged results for low dimensional simultion schemes are displayed for clean and contaminated structures, respectively. All averaged results for high dimesional simulation configurations, which are clean and contaminated, are shown in Table 6.9 and 6.10, respectively.

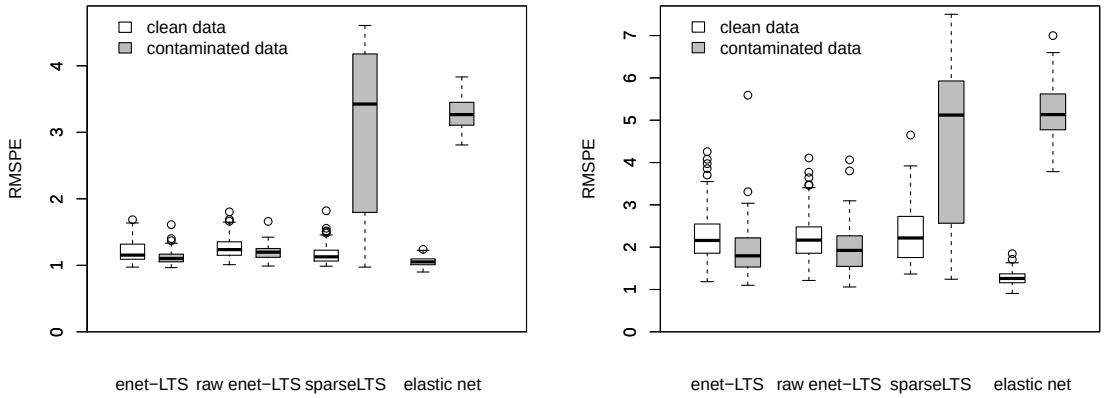

Figure 6.2 Root mean squared prediction error (RMSPE) for linear regression. Left: low dimensional data set ($n = 150$ and $p = 60$); right: high dimensional data set ($n = 50$ and $p = 100$).

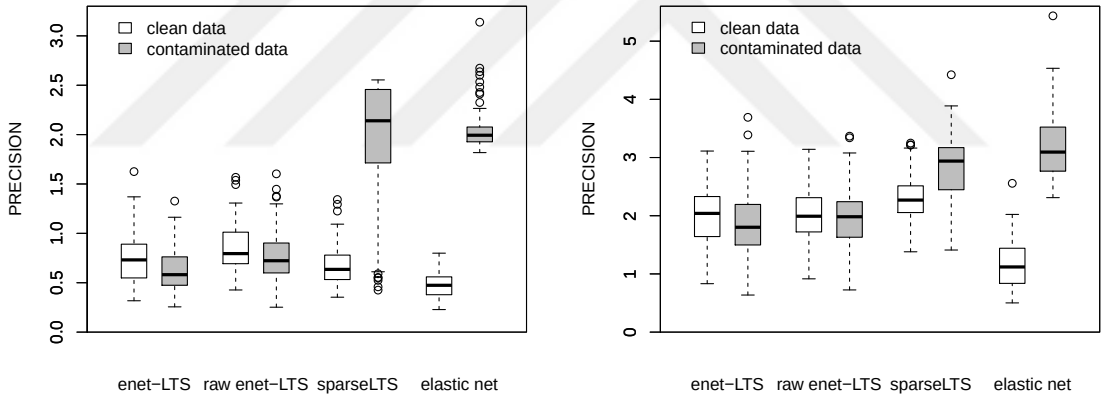

Figure 6.3 Precision of the estimators (PRECISION) for linear regression. Left: low dimensional data set ($n = 150$ and $p = 60$); right: high dimensional data set ($n = 50$ and $p = 100$).

Table 6.7 Results for low dimensional scheme for lienar regression ($n = 150$ and $p = 60$) with no contamimation: The root mean squared prediction error (RMSPE), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	No Contamination			
mean of:	RMSPE	Bias	FPR	FNR
enet-LTS	1.21	0.73	0.43	0.00
raw enet-LTS	1.28	0.86	0.50	0.00
sparseLTS	1.16	0.67	0.05	0.00
elastic net	1.06	0.47	0.21	0.00

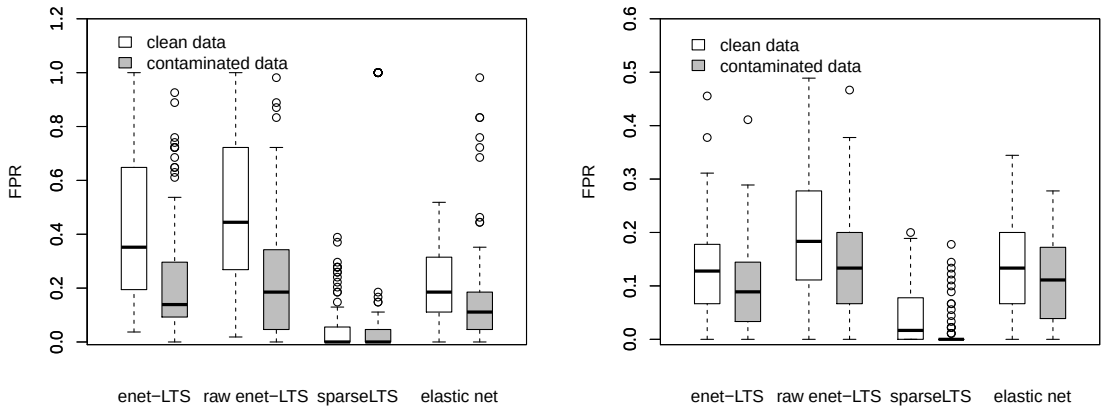

Figure 6.4 False positive rate (FPR) for linear regression. Left: low dimensional data set ($n = 150$ and $p = 60$); right: high dimensional data set ($n = 50$ and $p = 100$).

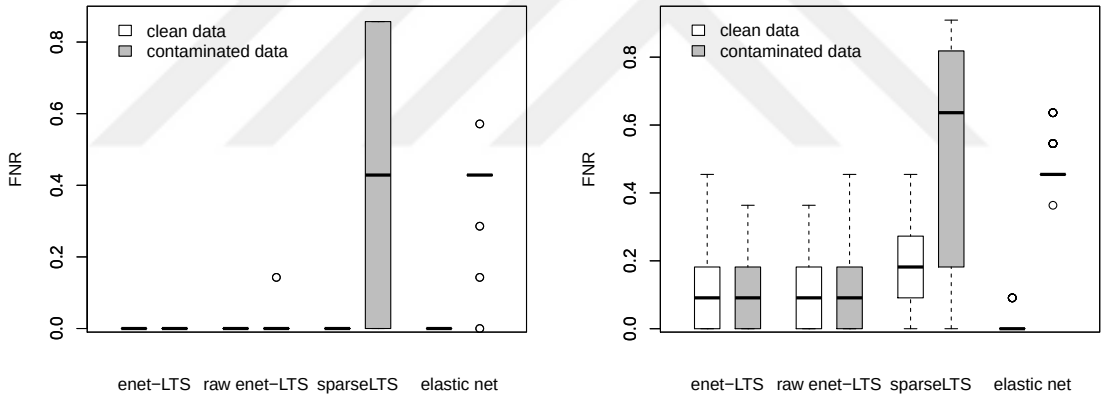

Figure 6.5 False negative rate (FNR) for linear regression. Left: low dimensional data set ($n = 150$ and $p = 60$); right: high dimensional data set ($n = 50$ and $p = 100$).

Table 6.8 Results for low dimensional scheme for lienar regression ($n = 150$ and $p = 60$) with contamination: The root mean squared prediction error (RMSPE), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	Contaminated			
mean of:	RMSPE	Bias	FPR	FNR
enet-LTS	1.12	0.63	0.23	0.00
raw enet-LTS	1.20	0.77	0.25	0.00
sparseLTS	3.04	1.97	0.18	0.42
elastic net	3.29	2.04	0.16	0.41

Table 6.9 Results for high dimensional scheme for lienar regression ($n = 50$ and $p = 100$) with no contamination: The root mean squared prediction error (RMSPE), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	No Contamination			
mean of:	RMSPE	Bias	FPR	FNR
enet-LTS	2.27	2.00	0.13	0.10
raw enet-LTS	2.25	2.01	0.20	0.09
sparseLTS	2.29	2.31	0.05	0.18
elastic net	1.28	1.18	0.14	0.01

Table 6.10 Results for high dimensional scheme for lienar regression ($n = 50$ and $p = 100$) with contamination: The root mean squared prediction error (RMSPE), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	Contaminated			
mean of:	RMSPE	Bias	FPR	FNR
enet-LTS	1.91	1.87	0.10	0.09
raw enet-LTS	1.97	1.97	0.15	0.10
sparseLTS	4.51	2.80	0.01	0.54
elastic net	5.17	3.18	0.11	0.48

6.3 Simulation Studies for Robust and Sparse Logistic Regression

6.3.1 Sampling schemes for robust and sparse logistic regression

We also consider two different scenarios for logistic regression, a "low dimensional" data set with $n = 150$ and $p = 50$ and a "high dimensional" data set with $n = 50$ and $p = 100$. The data matrix is $\mathbf{X} = (\mathbf{X}_a, \mathbf{X}_b)$, where $\mathbf{X}_a$ has the dimension $n \times p_a$ and $\mathbf{X}_b$ is of dimension $n \times p_b$, with $p = p_a + p_b$. The data matrices are generated independently from $\mathcal{N}_{p_a}(\mathbf{0}, \mathbf{\Sigma}_a)$ with $\mathbf{\Sigma}_a = \rho_a^{|j-k|}$, $1 \leq j, k \leq p_a$, and $\mathcal{N}_{p_b}(\mathbf{0}, \mathbf{\Sigma}_b)$ with $\mathbf{\Sigma}_b = \rho_b^{|j-k|}$, $1 \leq j, k \leq p_b$, respectively. While the first block consists of the informative variables with $p_a = 0.1p$, the second block represents uninformative variables with $p_b = 0.9p$. We take $\rho_a = 0.9$ for a high correlation among the informative variables, and $\rho_b = 0.5$ for moderate correlation among the uninformative variables.

The coefficient vector $\boldsymbol{\beta}$ consists of ones for the first 10% of the entries, and zeros for

the remaining uninformative block. The elements of the error term ε_i are generated independently from $\mathcal{N}(0,1)$. The grouping variable is then generated according to the model

$$y_i = \begin{cases} 0, & \text{if } 1+\mathbf{x}_i^T\boldsymbol{\beta}+\varepsilon_i \leq 0 \\ 1, & \text{if } 1+\mathbf{x}_i^T\boldsymbol{\beta}+\varepsilon_i > 0 \end{cases} \quad i = 1,2,\ldots,n. \tag{6.10}$$

With this setting, both groups are of approximately the same size.

Contamination is introduced by adding outliers only to the informative variables. Denote n_0 the number of observations in class 0. Then the first $\lfloor 0.1n_0 \rfloor$ observations of group 0 are replaced by values generated from $\mathcal{N}(20,1)$. In order to create "vertical" outliers in addition to leverage points, we assign those first $0.1n_0$ observations of class 0 a wrong class membership.

6.3.2 Performance Measures

The mean of the negative log-likelihoods or deviances (MNLL) for logistic regression,

$$\text{MNLL}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} d(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}}, y_i), \tag{6.11}$$

where y_i and $\mathbf{x}_i$, $i = 1,\ldots,n$, indicate the observations of the test data set, $\hat{\boldsymbol{\beta}}$ denotes the coefficient vector and $\hat{\beta}_0$ stands for the estimated intercept term obtained from the training data set. In logistic regression we also calculate the misclassification rate (MCR), defined as

$$\text{MCR} = \frac{m}{n} \tag{6.12}$$

where m is the number of misclassified observations from the test data after fitting the model on the training data. Similarly, we consider the precision of the coefficient estimate as a quality criterion, defined by Eq. (6.7). Additionally, the sparsity of the coefficient estimators is compared by FPR and the FNR defined as Eq. (6.8) and (6.9).

As mentioned before while the FPR is the proportion of non-informative variables that are incorrectly included in the model, the FNR is the proportion of informative variables that are incorrectly excluded from the model. A high FNR usually has a bad effect on the prediction performance since it inflates the variance of the estimator.

These evaluation measures are calculated for the generated data in each of 100 simulation replications separately, and then summarized by boxplots. The smaller the value for these criteria, the better the performance of the method.

6.3.3 Results for Robust and Sparse Logistic Regression

Figures 6.6–6.10 summarize the simulation results for logistic regression. As before, the left plots refer to the low dimensional case, and the right plots to the high dimensional data. Within one plot, the results for uncontaminated and contaminated data are directly compared. The misclassification rate in Figure 6.6 is around 10% for all methods, and it is slightly higher in the high dimensional situation. In case of contamination, however, this rate increases enormously for the classical method elastic net.

The average deviances in Figure 6.7 show that the reweighting of the enet-LTS estimator clearly improves the raw estimate in both the low and high dimensional cases. It can also be seen that elastic net is sensitive to the outliers. The precision of the parameter estimates in Figure 6.8 reveal a remarkable improvement for the reweighted enet-LTS estimator compared to the raw version, while there is not any clear effect of the contamination on the classical elastic net estimator.

The FPR in Figure 6.9 shows a certain difference between uncontaminated and contaminated data for the elastic net, but otherwise the results are quite comparable. A different picture is visible from the FNR in Figure 6.10, where especially in the low dimensional case the elastic net is very sensitive to the outliers. Overall we conclude that the enet-LTS performs very well in case of contamination even though this was not clearly visible in the precision, and it also yields reasonable results for clean data.

In Table 6.11 and 6.12, the averaged results for low dimensional simultion schemes are

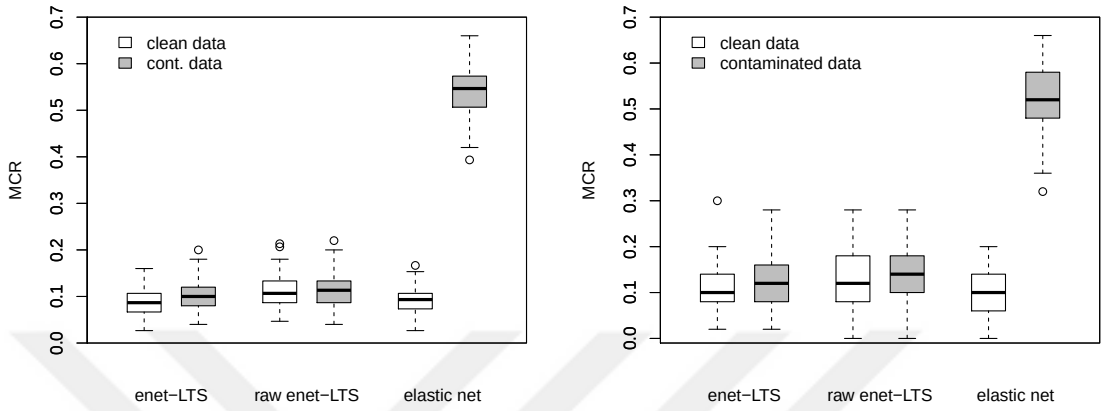

Figure 6.6 Misclassification rate for logistic regression. Left: low dimensional data set ($n = 150$ and $p = 50$); right: high dimensional data set ($n = 50$ and $p = 100$).

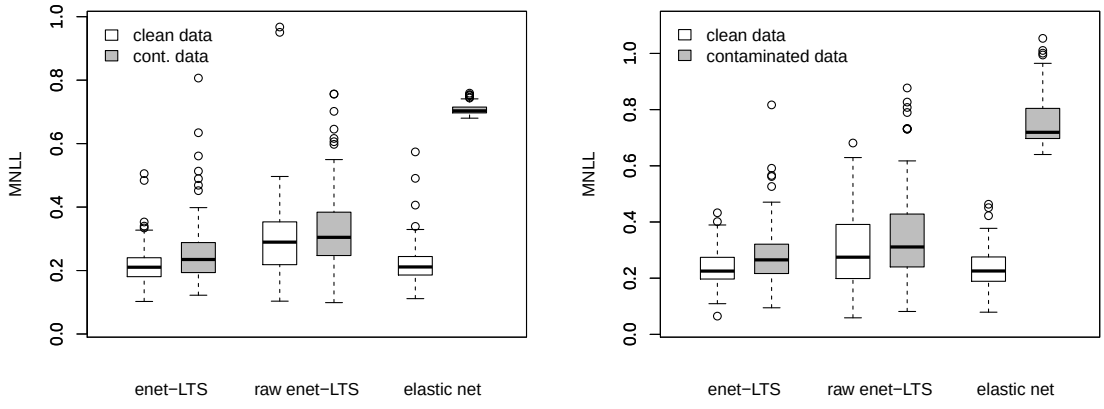

Figure 6.7 The mean of negative likelihood (MNLL) function for logistic regression. Left: low dimensional data set ($n = 150$ and $p = 50$); right: high dimensional data set ($n = 50$ and $p = 100$).

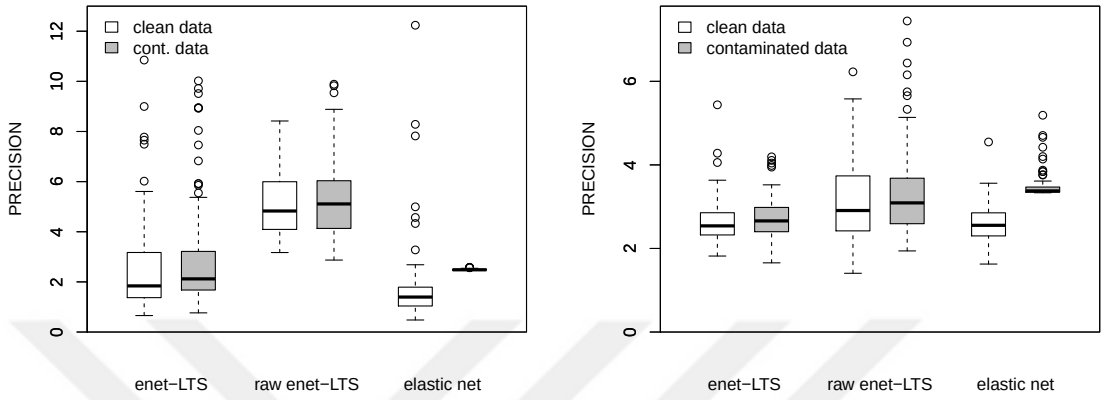

Figure 6.8 Precision of the estimators (PRECISION) for logistic regression. Left: low dimensional data set ($n = 150$ and $p = 50$); right: high dimensional data set ($n = 50$ and $p = 100$).

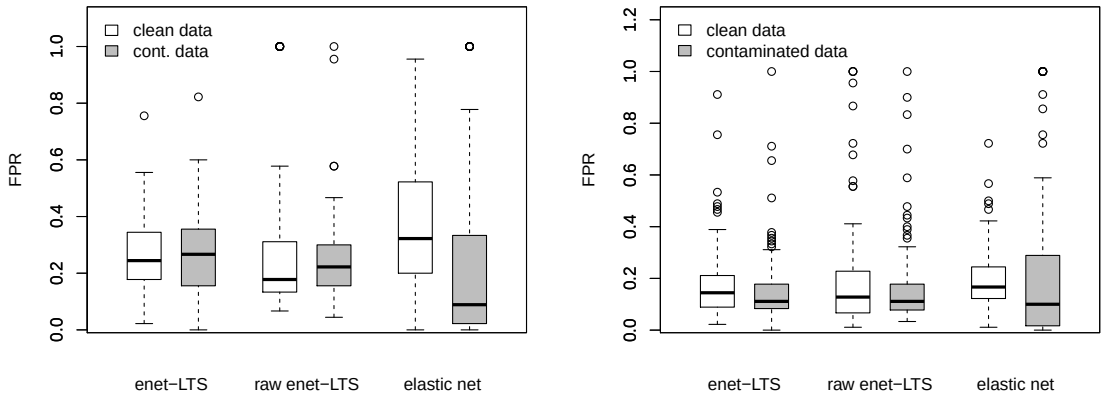

Figure 6.9 False positive rate (FPR) for logistic regression. Left: low dimensional data set ($n = 150$ and $p = 50$); right: high dimensional data set ($n = 50$ and $p = 100$).

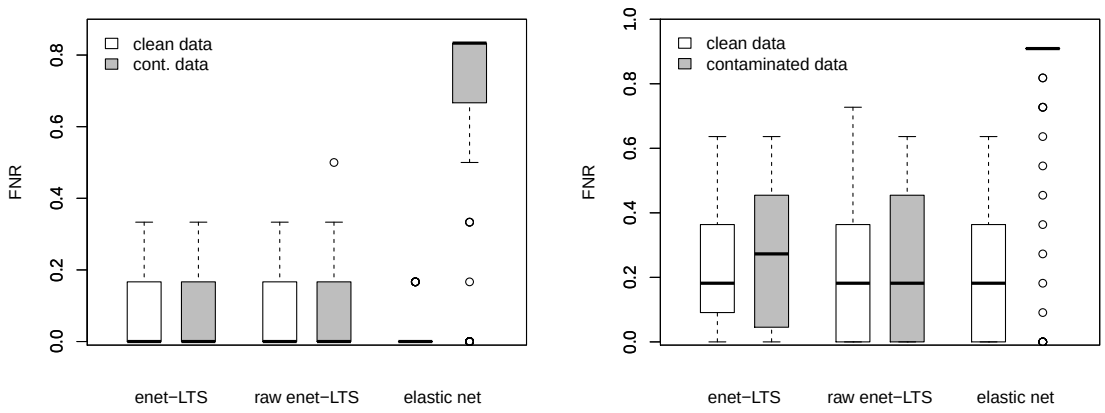

Figure 6.10 False negative rate (FNR) for logistic regression. Left: low dimensional data set ($n = 150$ and $p = 50$); right: high dimensional data set ($n = 50$ and $p = 100$).

displayed for clean and contaminated structures, respectively. All averaged results for high dimesional simulation settings, which are clean and contaminated, are shown in Table 6.13 and 6.14, respectively.

Table 6.11 Results for low dimensional scheme for logistic regression ($n = 150$ and $p = 50$) with no contamination: mean of negative log-likelihood (MNLL), the misclassification rate (MCR), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	No Contamination				
mean of:	MNLL	MCR	Bias	FPR	FNR
enet-LTS	0.22	0.09	2.52	0.27	0.05
raw enet-LTS	0.36	0.11	5.97	0.28	0.08
elastic net	0.22	0.09	1.75	0.37	0.01

Table 6.12 Results for low dimensional scheme for logistic regression ($n = 150$ and $p = 50$) with contamination: mean of negative log-likelihood (MNLL), the misclassification rate (MCR), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	Contaminated				
mean of:	MNLL	MCR	Bias	FPR	FNR
enet-LTS	0.10	0.10	2.96	0.27	0.06
raw enet-LTS	0.11	0.11	5.38	0.25	0.09
elastic net	0.54	0.54	2.49	0.23	0.68

Table 6.13 Results for high dimensional scheme for logistic regression ($n = 50$ and $p = 100$) with no contamination: mean of negative log-likelihood (MNLL), the misclassification rate (MCR), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	No Contamination				
mean of:	MNLL	MCR	Bias	FPR	FNR
enet-LTS	0.24	0.10	2.65	0.18	0.22
raw enet-LTS	0.34	0.13	3.48	0.20	0.21
elastic net	0.24	0.10	2.61	0.20	0.19

Table 6.14 Results for high dimensional scheme for logistic regression ($n = 50$ and $p = 100$) with contamination: mean of negative log-likelihood (MNLL), the misclassification rate (MCR), the bias of the estimators (Bias), the false positive rate (FPR) and the false negative rate (FNR), averaged over m=100 runs.

	Contaminated				
mean of:	MNLL	MCR	Bias	FPR	FNR
enet-LTS	0.28	0.12	2.75	0.16	0.25
raw enet-LTS	0.36	0.15	3.33	0.18	0.25
elastic net	0.80	0.52	3.49	0.23	0.79

CHAPTER 7

REAL DATA EXAMPLES

This chapter is dedicated to show the goodness of the proposed estimators on some different real data sets.

7.1 Real Data Example for Robust Linear Regression

In this section we focus on applications with linear regression. The first real data is taken to compare MM-Liu estimator with robust counterpart, LTS-Liu, and non-robust counterpart, LS-Liu. Then, the performance of PRM-Liu estimator is shown on a high dimensional data set. Model evaluation is done with 5-fold cross validation, i.e. each fold is used as test set once, a model is estimated on the training set, and the mean squared error is calculated for the test set. In these real data examples it is unknown if outliers are present. In order to avoid an influence of potential outliers on the evaluation of a model, the 10% trimmed mean suared error is calculated to compare the models.

7.1.1 Analysis of the Employment Data for Turkey

We consider a data set which was also used by [4] in the context of LTS-Liu regression as follows Table 7.1. The data contain macroeconomic variables of Turkey in the years 1988-2006. The independent variables are the non-institutionalized population for employment older than 14 years of age, the number of unemployed people, the year, and the gross national product (GNP). The dependent variable y represents the number of people employed in Turkey in this time period. It is known that this data set has both

multicollinearity problem and outliers [4].

Table 7.1 Employment data for Turkey.

Year	Employment	GNP($)	Unemployment	Population
1988	17,754	1684	1637	31,461
1989	18,222.5	1959	1712.5	31,948
1990	18,541.5	2682	1615.5	33,066
1991	19,294	2621	1725.5	34,248
1992	19,467	2708	1810	35,277
1993	18,506	3004	1821	36,153
1994	20,011	2184	1877	37,114
1995	20,594	2759	1704	38,115
1996	21,198	2928	1505	39,071
1997	21,207	3079	1557	40,020
1998	21,785	3255	1611	40,915
1999	22,056	2879	1836	41,809
2000	21,582	2965	1499	42,612
2001	21,525	2123	1969	43,455
2002	21,350	2598	2467	44,224
2003	21,146	3383	2492	44,974
2004	21,790	4172	2502	45,813
2005	22,046	5008	2519	46,620
2006	22,328	5477	2445	47,391

Figure 7.1 (left) shows the resulting MSE of the regression estimates, see equation (3.9) and the right plot shows the MSE of the response, see equation (3.10). Thus, in the left plot we get information on the quality of the parameter estimates (results are reported here for scaled data), while the right plot informs about the fit between actual data and the predicted model (here back-transformed to the original scale). Note that the value of MSE is computed with upper 10% trimming. The case $\lambda = 1$ corresponds to the results of the plug-in estimator (LS, LTS, and MM, respectively). All three Liu counterparts improve the MSE. One can see that the MM-Liu estimator leads to the best solution, followed by the LTS-Liu estimator. The LS-Liu estimator shows worst performance, in particular for the prediction.

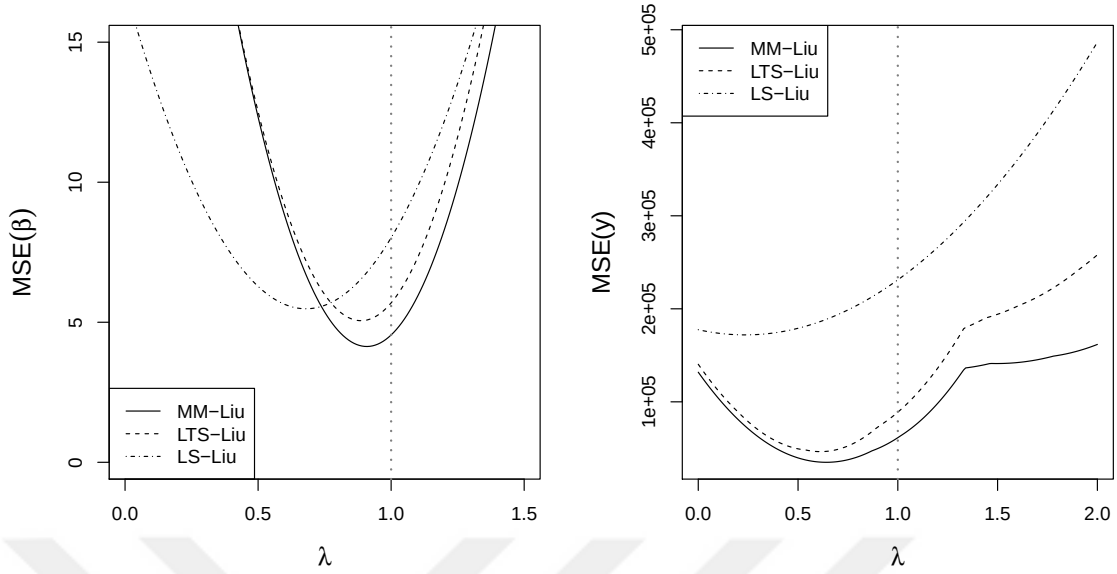

Figure 7.1 Results of the Liu-type estimators for the Turkish employment data. Left: results based on the MSE of the regression estimates; right: results based on minimizing the MSE of the prediction.

7.1.2 Analysis of the Glass Vessels Data

We consider a data set containing information on archaeological glass vessels from the 16^{th} and 17^{th} century that were excavated in Antwerp. The data set originates from [36], and the aim was to learn more about the production of these vessels, especially their origin and possible trade connections between known producers. The number of glass vessels is $n = 180$ and each of these glass vessels was analyzed by an electron-probe X-ray microanalysis (EXPMA) leading to 1920 characteristics. It is known from previous studies that the data contain outliers according to measurements with a different detector efficiency [22]. Following [32], we will only consider spectra in the range 15 to 500, because outside this range the frequencies are very low. The resulting data set has $n = 180$ observations and $p = 486$ variables which are highly collinear. The glass vessels have also been analyzed according to the concentration of chemical compounds. Here we focus on the oxide lead (PbO) which is used as response variable.

In Figure 7.2 we compare the resulting MSE of prediction for the PRM-Liu estimator, where the number of components ranges from 1 to 9. Here we employ the PRM-Liu estimator for the original values of PbO (left) and for the log-transformed PbO values (right), since the PbO values are heavily right-skewed. The optimal biasing parameter λ is

determined by 5-fold CV. One can see that log-transformation improved the MSE, and that a three-component model with $\lambda = 300$ gives the overall best MSE. Note that the result of the PRM estimator (without Liu) correspond to the values for $\lambda = 1$, and in case of the log-transformed response they can be clearly improved by the PRM-Liu estimator.

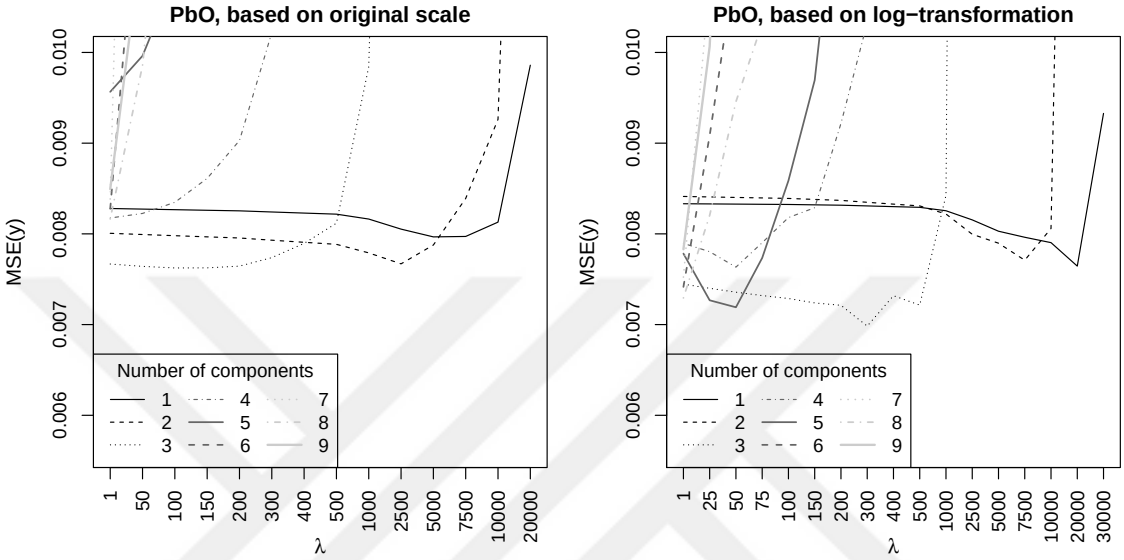

Figure 7.2 Glass vessel data: Resulting MSE of prediction based on CV for the PRM-Liu estimator, using the original (left) and the log-transformed (right) response PbO. The results change with the choice of the biasing parameter λ, and with the number of components.

7.2 Real Data Example for Robust and Sparse Linear Regression

In this section we give two examples for linear regression to compare the enet-LTS estimator and its raw version with non-robust elastic net estimator. The model selection is conducted as described in Section 4.2. Model evaluation is done with leave-one-out cross validation, i.e. each observation is used as test observation once, a model is estimated on the remaining observations, and the root mean squared prediction error is calculated for the test observation. Since in these real data examples it is unknown if outliers are present the 25% trimmed root mean squared prediction error is calculated to compare the models in order to avoid an influence of potential outliers on the evaluation of a model.

7.2.1 Analysis of the NCI Data

The performance of the enet-LTS estimator in case of linear regression is shown on the cancer data, which measures 60 human cancer cell lines, from the National Cancer Institute. Data can be downloaded from (http://discover.nci.nih.gov/cellminer/). Since the 40th observation has all missing values, it is out of calculation and $n = 59$. The gene expression data is obtained with an Affymetrix HG-U133A chip and normalized using the GCRMA method. The resulting data includes $p = 22.283$ predictors. Reverse-phase protein lysate arrays include the expression of 162 proteins. Instead of modeling the relationship for each protein expression seperately, we take the similar idea of Lee et al. [37] and Alfons et al. [13], namely, we choose one of the protein expression variables as predictand. Therefore, we focus on ADPRT–6 as predictand which corresponds to 4th of the protein expression variables.

Concerning prediction performance, the trimmed root mean squared prediction error (RMSPE) is computed via leave-one-out cross-validation (CV). The comparisons of raw and reweighted enet-LTS estimators with the classical elastic net estimator are reported in Table 7.2. According to trimmed RMSPE values, enet-LTS outperforms and is followed by raw enet-LTS. When we look at the number of variables of each method in Table 7.2, it seems that the selected tuning parameter α_{opt} is quite bigger for the classical elastic net than for the enet-LTS method. Since $\alpha_{opt} = 0.575$ for enet-LTS and 0.975 for elastic net, this claim is verified easily.

Figure 7.3 shows residuals of the classical elastic net model and reweighted elastic net model. While the classical elastic net estimator can not detect outliers, the proposed method does seccesfully. To support this result, the fitted values versus response variable are given in Figure 7.4, where (left) we can see that in some of the fitted values are very far away from the original response values and those correspond to outliers. On the other hand, they are not distinguisable in case of the classical elastic net model (right).

Table 7.2 NCI data: number of variables in the optimal models, and trimmed root mean squared prediction error from leave-one-out cross validation of the optimal models.

	number variables	trimmed RMSPE
enet-LTS	51	0.17190
raw enet-LTS	47	0.17600
elastic net	25	0.20664

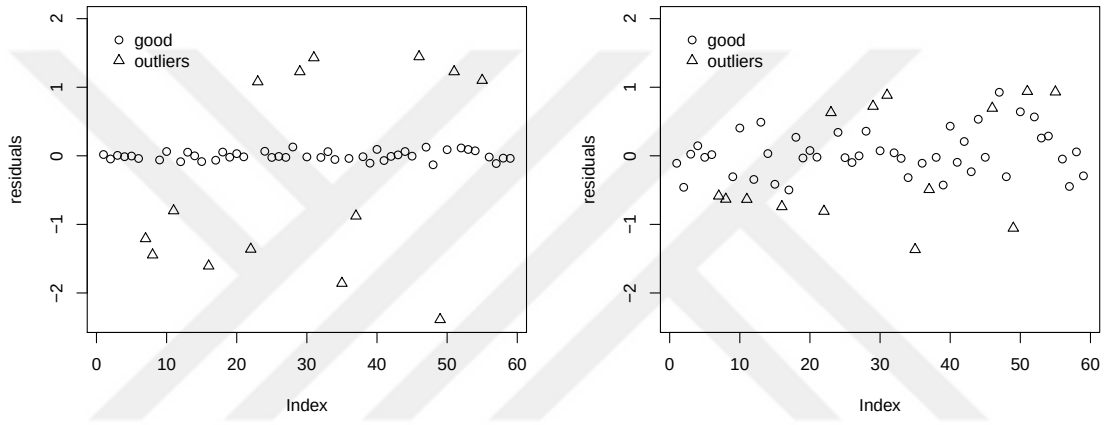

Figure 7.3 NCI data: Residuals of reweighted enet-LTS (left) and elastic net (right) estimators vs indexes which correspond to ordered observations on NCI Data.

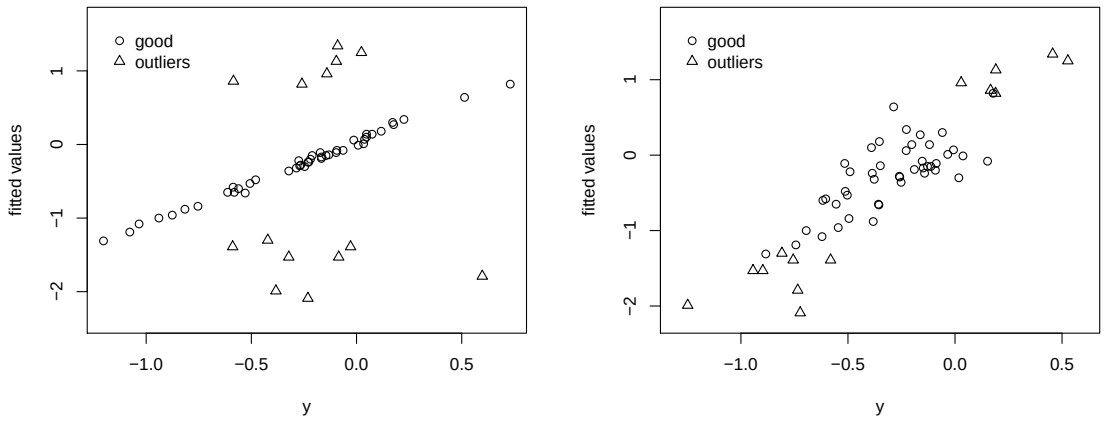

Figure 7.4 NCI data: Fitted values of reweighted enet-LTS (left) and elastic net (right) estimators vs response variable y.

7.2.2 Analysis of the Glass Vessels Data

As in section 7.1.2, we consider the archaeological glass vessels data, which is analyzed by [36], from the 16^{th} and 17^{th} century. The number of glass vessels is $n = 180$ and each of these glass vessels was analyzed by an electron-probe X-ray microanalysis (EXPMA) leading to p = 1920 spectra for each vessel. Similarly, we will only consider spectra in the range 15 to 500, which have highest frequences, instead of taking all variables of size $p = 1920$. Therefore the resulting data set has $n = 180$ observations and $p = 486$ variables which are highly collinear. As response variable, we focus on the oxide lead (PbO).

The quality of the selected models is summarized in Table 7.3. The trimmed root mean squared prediction error of the enet-LTS method is smaller than the elastic net. The reweighting step in enet-LTS leads the model with less variable as well as improving the model. Although both enet-LTS models include more variables than the elastic net model, this is not a big difference as seen in Table 7.3. While the penalty gives higher emphasis on the l_1 term with $\alpha_{opt} = 0.95$ for the elastic net model, it has a moderate value $\alpha_{opt} = 0.6$ for enet-LTS.

Figure 7.5 (left) shows residuals of the reweighted enet-LTS model. In this case, some of the resiuals are quite far away from the zero line, therefore corresponding observations to them are determined as outliers. As expected all observations behave very closely to each other in case of elastic net model Figure 7.5 (right). Further, it can be observed that neighboring variables, which are correlated, have similar coefficients. This is favored by the l_2 term in the elastic net penalty. In Figure 7.10 (right) the coefficient estimates of the elastic net model are visualized. Fewer coefficients are non-zero than for enet-LTS which was favored by the l_1 term in the elastic net penalty, but in the second block of non-zero coefficients neighboring variables receive very different coefficient estimates.

Table 7.3 Glass vessel data: number of variables in the optimal models, and trimmed root mean squared prediction error from leave-one-out cross validation of the optimal models.

	number variables	trimmed RMSPE
enet-LTS	43	0.00279
raw enet-LTS	61	0.00294
elastic net	39	0.00446

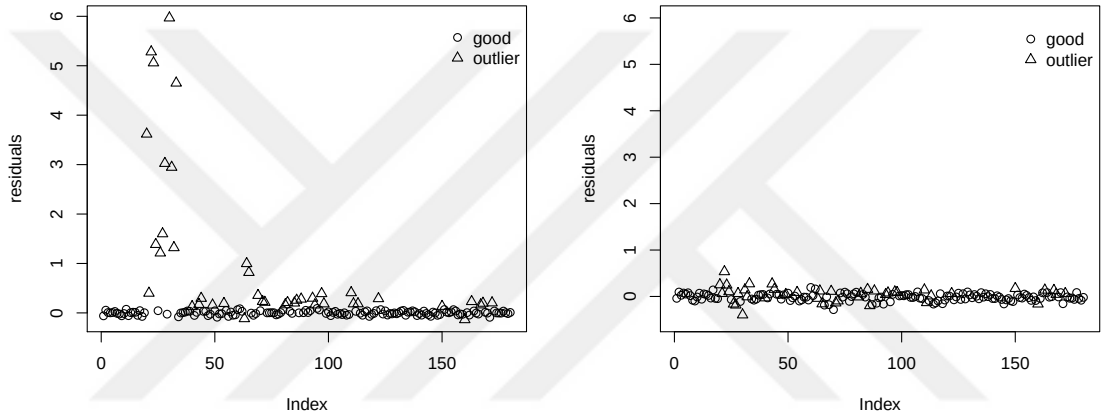

Figure 7.5 Glass vessel data: Residuals of reweighted enet-LTS (left) and elastic net (right) estimators vs indexes which correspond to ordered observations on Glass Vessels Data.

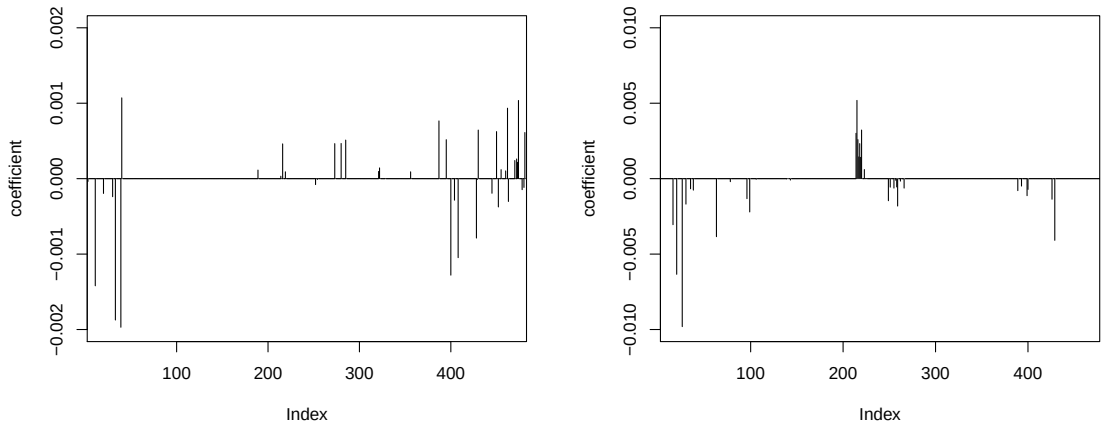

Figure 7.6 Glass vessel data: coefficient estimate of the reweighted enet-LTS (left) and coefficient estimate of the elastic net (right)

7.3 Real Data Example for Robust and Sparse Logistic Regression

In this section we focus on applications with logistic regression, and compare the non-robust elastic net estimator with the robust enet-LTS method. The model selection is conducted as described in Section 5.2. Model evaluation is done with leave-one-out cross validation, i.e. each observation is used as test observation once, a model is estimated on the remaining observations, and the negative log-likelihood is calculated for the test observation. In these real data examples it is unknown if outliers are present. In order to avoid an influence of potential outliers on the evaluation of a model, the 25% trimmed mean of the negative log-likelihoods is calculated to compare the models.

7.3.1 Analysis of the Meteorite Data

The time-of-flight secondary iron mass spectroscope COSIMA [38] was sent to the comet Churyumov-Gerasimenko in the Rosetta space mission by the ESA to analyze the elemental composition of comet particles which were collected there [39]. As reference measurements, samples of meteorites provided by the Natural History Museum Vienna were analyzed with the same type of spectroscope at Max Planck Institute for Solar System Research in Göttingen.

Here we apply our proposed method for logistic regression to the measurements from particles from the meteorites Ochansk and Renazzo with 160 and 110 spectra, respectively. We restrict the mass range to 1-100mu, consider only mass windows where inorganic and organic ions can be expected as described in [40] and variables with positive median absolute deviation. So we obtain $p = 1540$ variables. Further, the data is normalized to have constant row sum 100.

Table 7.4 summarizes the results for the comparison of the methods. The trimmed MNLL is much smaller for the enet-LTS estimator than for the classical elastic net method. The reweighting step improves the quality of the model further. The selected tuning parameter α_{opt} is much smaller for enet-LTS than for the classical elastic net method which strongly influences the number of variables in the models.

Table 7.4 Renazzo and Ochansk: Number of variables in the optimal models and trimmed mean negative log-likelihood from leave-one-out cross validation of the optimal models.

	number variables	trimmed MNLL
enet-LTS	397	0.00014
raw enet-LTS	294	0.00030
elastic net	136	0.00866

Figure 7.7 compares the Pearson residuals of the elastic net model and the enet-LTS model. In the classical approach no abnormal observations can be detected. With the enet-LTS model several observations are identified as outliers by the 1.25% and 98.25% quantiles of the standard normal distribution, which are marked as horizontal lines in Figure 7.7. Closer investigation showed that these spectra lie on the outer border of the measurement area and are potentially measured on the target instead of the meteorite particle. Their multivariate structure for those variables which are included in the model is visualized in Figure 7.8, where we can see that in some variables they have particularly large values compared to the majority of the group.

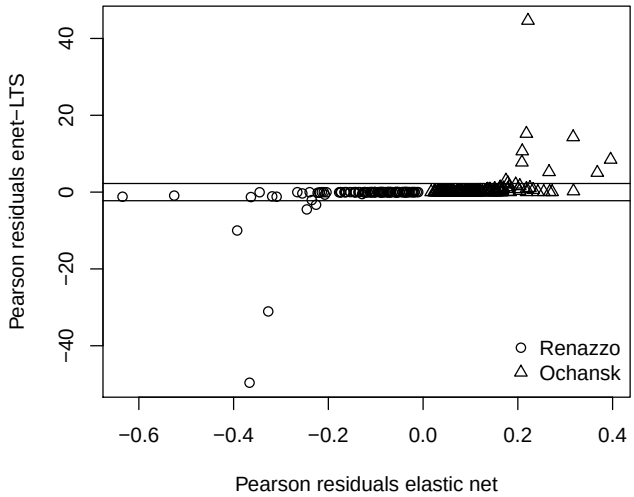

Figure 7.7 Renazzo and Ochansk: The Pearson residuals of elastic net and the raw enet-LTS estimator. The horizontal lines indicate the 0.0125 and the 0.9875 quantiles of the standard normal distribution.

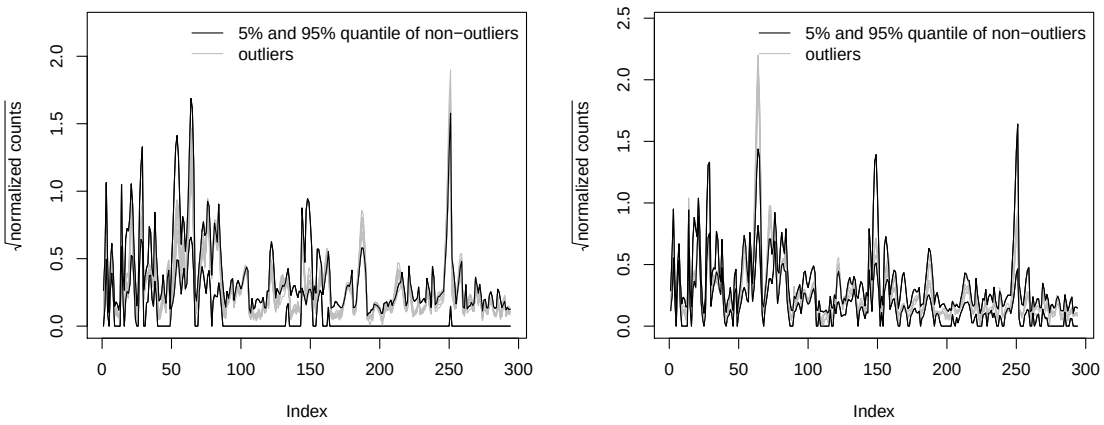

Figure 7.8 Renazzo and Ochansk: The index refers to the index of the variables included in the model of raw enet-LTS. The detected outliers are visualized by grey lines, while the black lines represent the 5% and 95% quantile of the non-outlying spectra for Ochansk (left) and Renazzo (right).

7.3.2 Analysis of the Glass Vessels Data

Archaeological glass vessels where analyzed with electron-probe X-ray micro-analysis to investigate the chemical concentrations of elements in order to learn more about their origin and the trade market at the time of their making in the 16^{th} and 17^{th} century [36]. Four different groups were identified, i.e. sodic, potassic, potasso-calcic and calcic glass vessels as seen in Figure 7.9. For demonstration of the performance of logistic regression, two groups are selected from the glass vessels data set. The first group is the potassic group with 15 spectra, the second group the potasso-calcic group with 10 spectra. As in [41] we remove variables with MAD equal to zero, resulting in $p = 1905$ variables.

The quality of the selected models is described in Table 7.5. The trimmed mean of the negative log likelihoods is much smaller for enet-LTS than for elastic net. The reweighting step in enet-LTS hardly improves the model, but includes more variables. Again, both enet-LTS models include more variables than the elastic net model. In the elastic net model the penalty gives higher emphasis on the l_1 term, i.e. $\alpha_{opt} = 0.8$; for enet-LTS it is $\alpha_{opt} = 0.05$.

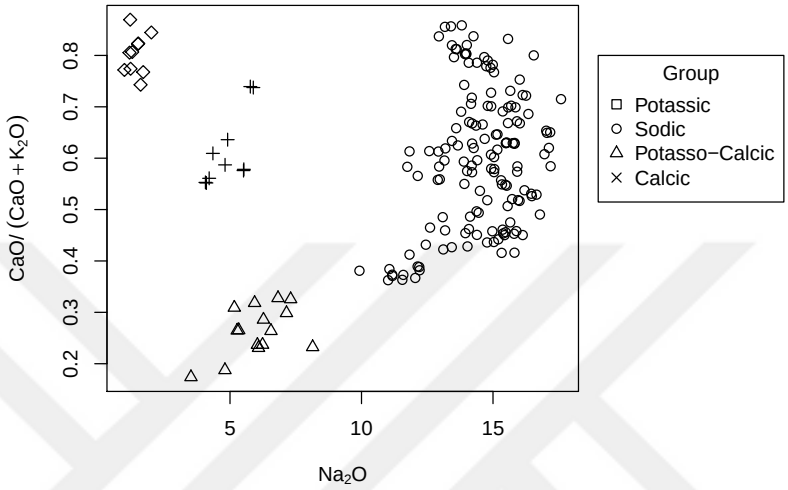

Figure 7.9 Glass vessels data visualisation: Ratio $\mathrm{CaO}/\left(\mathrm{CaO}+\mathrm{K_2O}\right)$ plotted against $\mathrm{Na_2O}$ concentration for all Glass vessels analyzed.

Table 7.5 Glass vessel data: number of variables in the optimal models, and trimmed mean negative log-likelihood from leave-one-out cross validation of the optimal models.

	number variables	trimmed MNLL
enet-LTS	448	0.000338
raw enet-LTS	375	0.000345
elastic net	50	0.004290

Different behavior of the coefficient estimates can be expected. Figure 7.10 (left) shows coefficients of the reweighted enet-LTS model corresponding to variables associated with potassium and calcium. The band which is associated with potassium has positive coefficients, i.e. high values of these variables correspond to the potassic group which is coded with ones in the response. High values of the variables in the band which is associated with calcium will favor a classification to the potasso-calcic group (coded with zero), since the coefficients for these variables are negative. Further, it can be observed that neighboring variables, which are correlated, have similar coefficients. This is favored by the l_2 term in the elastic net penalty. In Figure 7.10 (right) the coefficient estimates of the elastic net model are visualized. Fewer coefficients are non-zero than for enet-LTS which was favored by the l_1 term in the elastic net penalty, but in the second block of non-zero coefficients neighboring variables receive very different coefficient estimates.

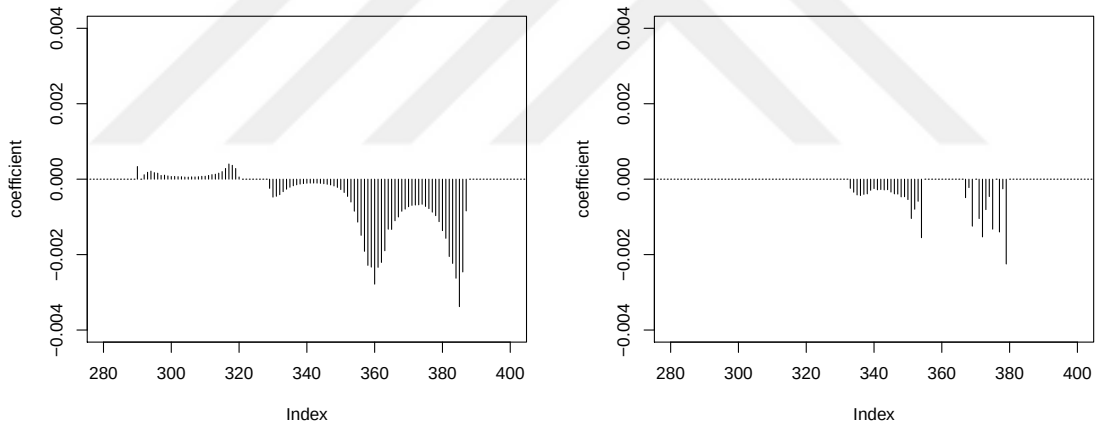

Figure 7.10 Glass vessels: coefficient estimate of the reweighted enet-LTS model (left) and coefficient estimate of the elastic net mode (right) for a selected variable range.

CHAPTER 8

COMPUTATION TIME

For our algorithm we employ the classical elastic net estimator as it is implemented in the R package *glmnet* [20]. So, it is natural to compare the computation time of our algorithm with this method. In the linear regression case we also compare with the sparse LTS estimator implemented in the R package *robustHD* [30]. For calculating the estimators we take a grid of five values for both tuning parameters α and λ. The data sets are simulated as in Chapter 6 for a fixed number of observations $n = 150$, but for a varying number of variables p in a range from 50 to 2000. In Figure 8.1 (left: linear regression, right: logistic regression), the CPU time is reported in seconds, as an average over 5 replications. In order to show the dependency on the number of observations n, we also simulated data sets for a fixed number of variables $p = 100$ with a varying number of observations $n = 50, 100, \ldots, 500$. The results for linear and logistic regression are summarized in Figure 8.2. The computations have been performed on an Intel Core 2 Q9650 @ 3000 GHz×4 processor.

Let us first consider the dependency of the computation time on the number of variables p for linear regression, shown in the left plot of Figure 8.1. Sparse LTS increases strongly with the number of variables p since it is based on the LARS algorithm which has a computational complexity of $\mathcal{O}(p^3 + np^2)$ [42]. Also for the smallest number of considered variables, the computation time is considerably higher than for the other two methods. The reason is that for each value of λ and each step in the CV the best subset is determined starting with 500 elemental subsets. In this setting at least 25,000 estimations of a Lasso model are needed, because for each cross validation step at each of the 5 values of λ, two C-steps for 500 elemental subsets are carried out, and for the 10 subsamples with lowest

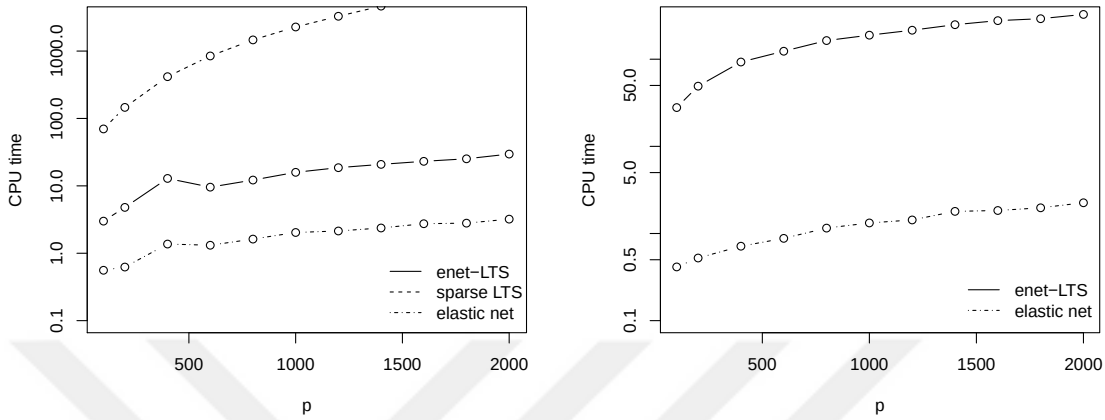

Figure 8.1 CPU time in seconds (log-scale), averaged over 5 replications, for fixed $n = 150$ and varying p; left: for linear regression; right: for logistic regression.

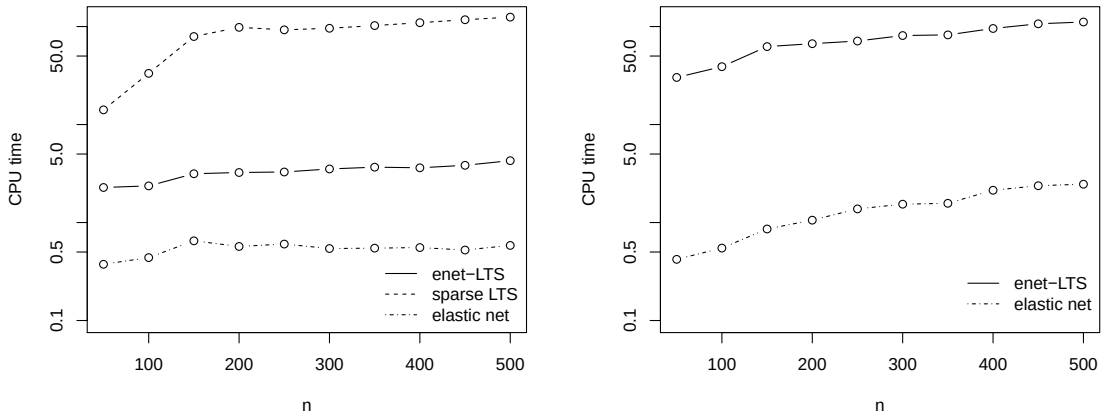

Figure 8.2 CPU time in seconds (log-scale), averaged over 5 replications, for fixed $p = 100$ and varying n; left: for linear regression; right: for logistic regression.

value of the objective function, further C-steps are performed. In contrast, the enet-LTS estimator starts with 500 elemental subsets only for one combination of α and λ, and takes the *warm start* strategy for subsequent combinations. This saves computation time, and there is indeed only a slight increase with p visible when compared to the elastic net estimator. In total approximately 1,700 elastic net models are estimated in this procedure, which are considerably fewer than for the sparse LTS approach. The computation time of sparse LTS also increases with n due to the computational complexity of LARS, while the increase is only minor for enet-LTS, see Figure 8.2 (left).

The results for the computation time in logistic regression are presented in Figure 8.1 (right) and 8.2 (right). Here we can only compare the classical elastic net estimator and the proposed robustified enet-LTS version. The difference in computation time between elastic net and enet-LTS is again due to the many calls of the `glmnet` function within enet-LTS. The robust estimator is considerably slower in logistic regression when compared to linear regression for the same number of explanatory variables or observations. The reason is that more C-steps are necessary to identify the optimal subset for each parameter combination of α and λ.

CHAPTER 9

RESULTS AND SUGGESTIONS

In this thesis, different robust methods for high dimensional data sets are introduced to solve the problems such as outliers in data, multicollinearity among the predictors. With this aim, first a fully robust version of the Liu estimator is proposed. The resulting MM-Liu estimator uses as a plug-in estimator the highly robust and efficient MM-estimator. Moreover, the definition of the Liu estimator was modified to include weights in order to downweight leverage points. This step makes also the selection of the biasing parameter robust, and it was not considered in the proposal of [4] who used the robust but inefficient LTS-estimator as a plug-in estimator. According to the simulation studies, the MM-Liu estimator outperforms the LTS-Liu estimator in all settings. In case of contamination, the MM-Liu estimator clearly outperforms the classical Liu estimator, but even without contamination the loss in performance is very low.

Afterwards, the idea of the robust Liu estimator is extended for the use with high-dimensional small sample size data. This estimator uses as a plug-in the PRM-estimator, a robustified partial least-squares (PLS) estimator [22]. Also in the definition of the resulting PRM-Liu estimator we used weights that result from the PRM-estimator for downweighting outliers. Although this concept does not really correspond to the philosophy of the Liu estimator, since the PRM-estimator is already applicable in case of multicollinearity, it could well be that "tuning" the PRM with an additional biasing parameter may lead to an improvement. Note that there is also another "tuned" version of the PLS-estimator, continuum regression, which allows to select between the "continuum" from LS-regression to principal component regression, with PLS as a special case [43]. The simulation study

has clearly demonstrated the advantage of PRM-Liu over the non-robust counterpart PLS-Liu in case of contamination, which is due to the robustness of the PRM-estimator. However, an advantage of PRM-Liu over PRM could not be seen in this setting. Only in the data example we could find an improvement of the PRM-estimator when using the PRM-Liu estimator. The example better accommodates the usability of PLS than the simulation setting, since PLS is assumed to perform well in case of an inherent latent structure.

Later on, concerning the another important thing with high dimensional data that they can include many uninformative variables which have no effect on the predictand or have very small contribution to the model, we not only focused on studying sparse estimation methods but also robust methods for linear and logistic regression with high dimensional. While robustness has been achieved by using trimming, sparsity is provided using the elastic net penalty. Therefore, this penalty allows for variable selection, can deal with high multicollinearity among the variables, and is thus very appropriate in high dimensional sparse settings. However, the idea of trimming usually leads to a loss in efficiency, and therefore a reweighting step was introduced. Overall, the outlined algorithms for linear and logistic regression turned out to yield good performance in different simulation settings, but also with respect to computation time. Particularly, it was shown that the idea of using "warm starts" for parameter tuning allows to save computation time, while the precision is still preserved. A competing method for robust high dimensional linear regression, the sparse LTS estimator [30], does not use this idea, and is thus much less attractive concerning computation time, especially in case of many explanatory variables. We should also admit that for other simulation settings (not shown here), the difference between sparse LTS and the enet-LTS estimator is not so big, or even marginal, depending on the exact setting.

For this reason, a further focus was on the robust high dimensional logistic regression case. We consider such a method as highly relevant, since in many modern applications in chemometrics or bio-informatics, one is confronted with data information from two groups, with the task to find a classification rule and to identify marker variables which support the rules. Outliers in the data are frequently a problem, and they can affect the

identification of the marker variables as well as the performance of the classifier. For this reason it is desirable to treat outliers appropriately. It was shown in simulation studies as well as in data examples, that in presence of outliers the new proposal still works well, while its classical non-robust counterpart can lead to poor performance.

The algorithms to compute the proposed estimators are implemented in R functions. The basis for the computation of the robust estimator is the R package *glmnet* [20]. This package also implements the case of multinomial and Poisson regression. Naturally, a further extension of the algorithms introduced here could go into these directions. Further work will be devoted to the theoretical properties of the family of enet-LTS estimators.

REFERENCES

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

[2] Liu, K., (1993). "A new class of biased estimates in linear regression", Communications in Statistics: Theory and Methods, 22:393-402.

[3] Arslan, O., and Billor, N., (2000). "Robust liu estimator for regression based on M-estimator", Journal of Applied Statistics, 27:39-47.

[4] Kan, B., Alpu, O. and Yazıcı B., (2013). "Robust ridge and robust liu estimator for regression based on the lts estimator", Journal of Applied Statistics, 1:799-821.

[5] Varmuza, K. and Filzmoser, P., (2008). Introduction to Multivariate Statistical Analysis in Chemometrics, Taylor and Francis, New York.

[6] Geladi, P. and Esbensen, K., (1990). "The start and early history of chemometrics: Selected interviews, part 1", Journal of Chemometrics, 4:337-354.

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

[8] Zou, H. and Hastie, T., (2005). "Regularization and variable selection via the elastic net", Journal of the Royal Statistical Society: Series B, 67(2):301-320.

[9] Filzmoser, P., Gshwandtner, M. and Todorov, V., (2012). "Review of sparse methods in regression and classification with application to chemometrics", Journal of Chemometrics, 26(3-4):42-51.

[10] Maronna, R.A., Martin, R.D. and Yohai, V.J., (2006). Robust Statistics: Theory and Methods, Wiley, New York.

[11] Rousseeuw, P.J. and Leroy, A.M., (2003). Robust Regression and Outlier Detection, Wiley, Second Edition: John Wiley and Sons, New York.

[12] Rousseeuw, P.J. and Van Driessen, K., (2006). "Computing LTS regression for large data sets", Data Mining and Knowledge Discovery, 12(1):29-45.

[13] 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.

[14] Öllerer, V., (2015). Robust and sparse estimation in high dimensions, PhD Thesis, KU Leuven, Belgium, KU Leuven Fakulteit Economie en Bedrijfswetenschappen.

[15] Darwish, K. and Büyüklü, A.H., (2015). "Robust linear regression using L1 penalized MM-estimation for high dimensionak data", American Journal of Theoretical Applied Statistics, 4(3):78-84.

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

[17] Croux, C., Dhaene, G. and Hoorelbeke, D., (2003). "Robust standard errors for robust estimators", Discussion Papers Series 03.16, KU Leuven, Belgium, 24:118-173.

[18] Bianco, V.J. and Yohai, A.M., (1996). Robust Estimation in Logistic Regression Model, in Robust Statistics, Data Analysis, and Computer Intensive Methods, 17-34; Lecture Notesin Statistics 109. Springer Verlag. Ed. H., Rieder: New York.

[19] Albert, A. and Anderson, J.A., (1984). "On the existence of maximum likelihood estimates in logistic regression models", Biometrika, 71:1-10.

[20] Friedman, J., Hastie, T., Simon, N. and Tibshirani, R. (2016). "glmnet: Lasso and Elastic Net Regularized Generalized Linear Models", R Foundation for Statistical Computing, Vienna, Austria, URL http://CRAN.R-project.org/package=glmnet. R package version 2.0-5.

[21] Park, H. and Konishi, S., (2016). “Robust logistic regression modeling via the elastic net-type regularization and tuning parameter selection”, Journal of Statistical Computation and Simulation, 86(7):1450-1461.

[22] Serneels, S., Croux, C., Filzmoser, P. and Espen, P.J.V., (2005). “Partial Robust M-Regression”, Chemometrics and Intelligent Laboratory Systems, 79:55-64.

[23] Donoho, D.L. and Huber, P.J., (1983). The notion of breakdown point, Festschrift for Erich L. Lehmann, P.J. Bickel, K.A. Doksum and J.L. Hodges, (eds), 157-184. Belmont, CA: Wadsworth.

[24] Hampel, F.R., “The influence curve and its role in robust estimation”, The Annals of Statistics, 69:383-393.

[25] Liang, Y.Z. and Kvalheim, O. (1996). “Robust methods for multivariate analysis – a tutorial review”, Chemometrics and Intelligent Laboratory Systems, 32(1):1-10.

[26] Liang, K.T., and Fang, Y.Z., (1996). “Robust multivariate calibration algorithm based on least median of squares and sequential number theory optimization method”, Analyst, 121(8):1025-1029.

[27] Rousseeuw, P.J., (1984). “Least median of squares regression”, Journal of American Statistical Association, 79(388):871-880.

[28] Yohai, V.J., (1987). “High breakdown point and high efficiency robust estimates for regression”, The Annals of Statistics, 15:642-656.

[29] Wold, H., (1973). Nonlinear Iterative Partial Least Squares (nipals) Modelling. In P.R. Krishnaiah, editor, Multivariate Analysis III, Academic Press, New York.

[30] Alfons, A., (2013). “Robust methods for high dimensional data”, R Foundation for Statistical Computing, Vienna, Austia. URL http://CRAN.R-project.org/package=robustHD. Rpackage version 0.4.0.

[31] McCullagh, P.Y. and Nelder, J.A., (1983). Generalized Linear Models, Chapman and Hall, New York.

[32] Maronna, R.A., (2011). “Robust ridge regression for high dimensional data”, Techometrics, 53:44-53.

[33] Filzmoser, P. and Kurnaz, F.S., (2017). “A new robust liu type estimator”, Communications in Statistics: Simulations and Computations, 2017.

[34] Khan, J.A., Van Aelst, S. and Zamar, R.H., (2007). “Robust linear model selection based on least angle regression”, J. Amer. Statist. Assoc., 102:1289-1299.

[35] Croux, C. and Haesbroeck, G., (2003). “Implementing the Bianco and Yohai estimator for logistic regression”, Computational Statistics and Data Analysis, 44(1-2):273-295.

[36] Janssens, K., Deraedt, I., Freddy, A. and Veekman, J., (1998). “Composition of 15 and 16 th century archeological glass vessels excavated in antwerp, belgium”, Mikrochima Acta, 15:253-267.

[37] Lee, D., Lee, W., Lee, Y. and Pawitan, Y., (2011). “Sparse partial least squares regression and its applications to high-throughout data analysis”, Chemometrics and Intelligent Laboratory Systems, 109:1-8.

[38] Kissel, J., Altwegg, K., Clark, B.C., Colangeli, L., Cottin, H., Czempiel, S., Eibl, J., Engrand, C., Fehringer, H.M., Feuerbacher, B. and et al., (2007). “COSIMA-high resolution time-of-flight secondry ion mass spectrometer for the analysis of cometary dust particles onboard Rosetta”, Space Science Reviews, 128(1-4):823-867.

[39] Schulz, R., Hilchenbach, M., Langevin, Y., Kissel, J., Silen, J., Briois, C., Engrand, C., Hornung, K., Baklouti, D., Bardyn, A. and et al. (2015). “Comet 67P/Churyumov-Gerasimenko sheds dust coat accumulated over the past four years”, Nature, 518(7538):216-218.

[40] Varmuza, K., Engrand, C., Filzmoser, P., Hilchenbach, M., Kissel, J., Kruger, H., Silen, J. and Trieloff, M., (2011). “Random projection for dimensionality reduction - applied to time-of-flight secondary ion mass spectrometry data”, Analtica Chimica Acta, 705(1):48-55.

[41] Filzmoser, P., Maronna, R. and Werner, M., (2008). “Outlier identification in high dimensions”, Computational Statistics and Data Analysis, 52(3):1694-1711.

[42] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R., (2004). “Least angle regression”, The Annals of Statistics, 32(2):407-499.

[43] Stone, M. and Brooks, R.J., (1990). “Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares and principal components regression”, Journla of the Royal Statistical Association: Series B (Methodological), 52:237-269.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname	: Fatma Sevinç KURNAZ
Date of birth and place	: 01.08.1986
Foreign Languages	: English
E-mail	: fskurnaz@gmail.com

EDUCATION

Degree	**Department**	**University**	**Date of Graduation**
PhD	Statistics	Yildiz Technical	19.06.2017
Master	Mathematics	Istanbul	07.07.2011
Undergraduate	Mathematics	Çanakkale Onsekiz Mart	16.06.2007

WORK EXPERIENCES

Year	**Institute**	**Enrollment**
2014–present	Yildiz Technical University,	Depart. of Statistics
2012–2014	Karadeniz Technical university,	Depart. of Stat. and Comp. Sciences

PUBLISHMENTS

Articles

1. Kurnaz F.S. and Akay K.U., Matrix Mean Squared Error Comparisons of Some Biased Estimators with Two Biasing Parameters, *Communications in Statistics–Theory and Methods*, 2017, DOI: 10.1080/03610926.2017.1335415. **(SCI-E)**

2. Filzmoser P., Kurnaz F. S., A Robust Liu Regression Estimator, *Communications in Statistics: Simulations and Computations*, 2017, DOI: 10.1080/03610918.2016.1271889. **(SCI-E)**

3. Kurnaz F. S., Akay K. U., A New Liu–type Estimator, *Statistical Papers*, 2014, DOI: 10.1007/s00362-014-0594-6. **(SCI-E)**

Conference Papers

1. **LinStat2014**, (Invited Speaker) Kurnaz F. S., Filzmoser P., A Robust Liu Regression Estimator, Linstat2014, International Conference on Trends and Perspectives in Linear Statistical Inference, Linköping, SWEDEN, (24-28 August 2014).

2. **LinStat2012**, Kurnaz F. S., Akay K. U., A New Liu-type estimator, International Conference on Trends and Perspectives in Linear Statistical Inference, Bedlewo, Poznan, POLAND, (16-20 July 2012).

3. **ICEOS2012**, Kurnaz F. S., Akay K. U., The Comparisons of some Biased Estimators Include the Two Biasing Parameters, 13'th International Conference of Econometrics, Operations Research and Statistics, Famagusta, KKTC, (24-26 May 2012).

Awards

1. Research Fellowship 2214/A for PhD Students by the Scientific and Technological Research Council of Turkey (TUBITAK), 2016.

2. The Young Scientists Award 3th, LinStat12, Bedlewo, Poznan, Poland, (2012).

3. Best ranking student among 2007 graduates of Mathematics Departments of Çanakkale Onsekiz Mart University, (2007).