TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

DISSERTATION

Sparse and robust modeling for high-dimensional data

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

unter der Leitung von
Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser,
Institut für Stochastik und Wirtschaftsmathematik (E105)

eingereicht an der Technischen Universität Wien an der Fakultät für Mathematik und Geoinformation von

Dipl.-Ing. Irene Hoffmann

Matrikelnummer 0825098

Diese Dissertation haben begutachtet:

Peter Filzmoser	Tim Verdonck	Beata Walczak
TU Wien	KU Leuven	University of Silesia

Wien, 10. Oktober 2017

Irene Hoffmann

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Irene Hoffmann
Attemsgasse 39/1/13
1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit einschließlich Tabellen, Karten und Abbildungen, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. Oktober 2017

Irene Hoffmann

Acknowledgements

First of all, I would like to thank my supervisor Prof. Peter Filzmoser who encouraged me and supported me in many ways. Thank you for all the opportunities and the contacts you provided me with, for your advise and many fruitful discussions.

I would also like to thank Prof. Varmuza for the insight he gave me into chemometrics and spectroscopy and for the many discussions on data preprocessing. Many thanks also to Sven Serneels and Prof. Christophe Croux. I learned many things from you and I highly appreciate the opportunity of working together.

My special thanks go to my friends and colleagues who shared not only an office with me, but also many challenging and joyful moments. Thank you to all my friends and especially to my family for encouraging words, for trying to understand my research and for practical support.

Finally, I express my gratitude to the Austrian Science Fund (FWF) for financial support through the project P 26871-N20.

Kurzfassung

Ein wichtiger Fokus in der Entwicklung neuer statistischer Methoden liegt seit einigen Jahren auf der Analyse hochdimmensionaler Daten. Klassische Regressions- und Klassifikationsmethoden benötigen Datenmatrizen mit vollem Rang, die mehr Beobachtungen als Variablen beinhalten. In vielen Anwendungsgebieten (z.B. der Bioinformatik oder Chemometrie) kann diese Anforderung aus praktischen Gründen nicht erfüllt werden. *Sparse modeling* umfasst eine Klasse von Methoden, die durch einen Strafterm Nullwerte bei der Koeffizientenschätzung bevorzugen und dadurch intrinsisch Variablen selektieren.

Eine weitere Herrausforderung in vielen Anwendungsgebieten sind Ausreißer in den Daten. Als Ausreißer werden Beobachtungen bezeichnet, die nicht der Struktur oder dem Trend der Mehrheit der Daten entsprechen und dadurch die Verteilungsannahmen klassischer Methoden verletzen und Modellschätzungen verzerren. Robuste Methoden beschränken den Einfluss extremer Werte auf die Modellschätzung und liefern stabile Modelle.

Diese Arbeit befasst sich mit robusten Methoden, die *sparse modeling* Ansätze integrieren und dadurch anwendbar auf hochdimensionale Daten sind. *Sparse partial robust M regression* ist eine robuste Methode, die partielle kleinste Quadrate Regression mit *sparse modeling* verbindet. Die latenten Variablen eines niedrigdimensionalen Raumes werden aus Linearkombinationen einer Teilmenge der originalen Variablen erzeugt. Mit den latenten Variablen wird ein robustes Regressionsmodell erzeugt. Die Methode wird für binäre Klassifikationsprobleme erweitert. *Robust sparse optimal scoring* ist eine weitere robuste Klassifikationsmethode, die auch auf Mehrgruppenprobleme angewandt werden kann und auf *least trimmed squares regression* basiert. Zuletzt werden zwei robuste Methoden vorgestellt, die durch einen *elastic net* Strafterm sowohl Variablenselektion integrieren als auch regularisierend auf die Koeffizienten wirken, wenn Variablen stark korrelieren.

Abstract

The development of statistical methods for high-dimensional data has become an important focus in recent research. Classical regression and classification approaches require full rank data matrices, with more observations than variables. In many areas of application (e.g. bioinformatics and chemometrics) this assumption is not met. Sparse methods describe a class of approaches where a penalty is imposed on the coefficient estimate to favour exact zero values and so intrinsically perform variable selection.

Another challenge in many applications are outliers in the data, which are observations that do not follow the structure of the majority of the data and so violate the distribution assumptions which are necessary for classical model estimation. Robust methods give stable estimates when outliers are present and model the relationship of the majority of the data.

The focus of this thesis is on the development of regression and classification methods, which are appropriate for high-dimensional data and data with outliers. Sparse partial robust M regression is a robust and sparse regression method. A robust subspace is identified, including only a subset of the original variables, where a robust regression model is estimated. This approach is then extended to binary classification problems. With the help of the optimal scoring approach, regression methods can be applied to classification problems. Robust sparse optimal scoring is a classification method based on least trimmed squares regression. Finally, sparse and robust linear regression and logistic regression methods are introduced based on least trimmed squares with an elastic net penalty, which induces sparsity and at the same time favours similar coefficient estimates for highly correlated variables.

Contents

CHAPTER 1

Introduction

Robust and sparse modeling addresses two common problems in data analysis. The first problem are outliers in the data, which are observations deviating from the pattern of the majority. Outliers can be contaminated measurements or other suspicious observations which one wants to exclude from the analysis or they are like the needle in a haystack which the analyst is looking for. Identifying deviating observations can be used to obtain warnings of production failures, in fraud detection or to detect uncommon changes in the picture frames of a survey camera. Robust models provide a description of the data based only on a majority, which is not distorted by few deviating observations.

Another challenge in many research areas is the steadily increasing amount of variables, which are measured in the experiments. Often it is assumed that only a small subset of variables measures characteristics which are of interest for the experiment. Then the challenge is to identify these variables of interest. Sparse modeling reduces the number of variables which are included in the model by simultaneous variable selection and model estimation due to a penalty term in the objective function of the model estimation.

1.1 Robust modeling

In statistical model estimation relationships between variables are derived from observed data generally with the aim to gain understanding of the connection between the variables and/or to predict unknown outcomes. Ignoring outliers in this process can induce severe problems for both aims. A single outlying observation can distort the estimated

relationship between variables and predictions can become completely unreliable. Therefore, robust methods, i.e. methods which are insensitive to a small fraction of observations which have extreme values, are an essential part of a statistician's tool kit. The underlying idea is to model the trend of the majority of the data.

Several approaches for robust regression analysis have been developed. Least trimmed squares (LTS) regression (Rousseeuw, 1984) minimizes a trimmed sum over the squared residuals, excluding the largest $\alpha\%$ of the squared residuals from the model estimation. A fast algorithm exists (Rousseeuw and Van Driessen, 2006) making it a popular robust alternative to least squares regression. The choice of α is a trade-off between robustness (the tolerated fraction of outliers in the data) and statistical efficiency.

Another robust regression method is the MM-estimator (Yohai, 1987). Starting with a highly robust coefficient estimate with low efficiency, its efficiency is improved by iterative reweighting steps. For the reweighting, weights between zero and one are assigned to each observation based on the size of the standardized residuals. Large absolute residuals result in low weights, which reduces the influence of the observation on the model estimation. Thus, a highly robust and efficient regression model is obtained.

In linear discriminant analysis (LDA) it is assumed that the data origins from g groups, which have the same covariance structure. A pooled sample covariance estimate is used to describe the within-group covariance structure. For robust classification models it is therefore crucial to obtain robust center and scatter matrices (Duda et al., 2012; McLachlan, 2004). The minimum covariance determinant (MCD) estimator (Rousseeuw, 1985; Rousseeuw and Driessen, 1999) is a popular choice for robust LDA (Hubert and Van Driessen, 2004; Todorov and Pires, 2007).

For a two group classification problem logistic regression is an alternative to LDA. Robust logistic regression was introduced by Bianco and Yohai (1996) and further development of the methodology presented in Croux and Haesbroeck (2003). All these robust methods, same as their classical counterparts, are limited to applications where more observations than variables are available.

1.2 Modeling with high-dimensional data

Many statistical methods assume a full rank data matrix, an assumption which is violated if the number of variables exceeds the number of observations. Here we refer to data sets with more variables than observations as *high-dimensional data*. With increasing number of variables the data space quickly gets more and more empty, which is often referred to

as the *curse of dimensionality* (see e.g. Beyer et al., 1999; Bennett et al., 1999). A large number of observations is necessary to adequately describe the full space, which is often unfeasible due to the limited number of observations (cost factor or other limits). The manifold hypothesis states that the observations are not spread in the whole data space, but located on a low dimensional manifold within the data space. This hypothesis is the basis for dimension reduction techniques and sparse modeling.

Dimension reduction

Under the assumption that variance equals information, the principal component analysis (PCA) is a well established method for dimension reduction. It is considered an unsupervised method, since it is applied only to the predictor matrix. In regression or classification problems, where the target is to model the response or the class-membership, the directions of highest variance of the predictors do not necessarily represent the most useful information to achieve this target. Nevertheless, PCA is often applied to predictor data before regression or discriminant analysis in practice.

Wold (1965) introduced the idea of partial least squares (PLS) regression. It is a supervised dimension reduction and model estimation technique. First, latent variables are obtained as linear combinations of the original variables such that they maximize the covariance to the response vector. Second, a linear regression model is estimated on the latent variables. The latent variables are constructed to be uncorrelated, so a stable regression estimator is obtained even for data sets with high multicollinearity.

Also for linear discriminant analysis, PLS is an appropriate method. The second step from PLS regression can be replaced by estimating an LDA model. It has been shown that the PLS discriminant model maximizes the between group covariance (Barker and Rayens, 2003).

Another approach to directly estimate a regression model based on high-dimensional data is the Ridge estimator (Hoerl and Kennard, 1970). The model estimation is restricted by a value the L_2 norm on the coefficients is restricted. This leads to continuous shrinkage of the coefficient estimates, adds a bias to the coefficients, but also reduces the variance. With a proper restriction of the L_2 norm the model precision can be improved greatly by this bias-variance trade-off.

Sparse modeling

Sparse modeling combines model estimation with intrinsic variable selection. The underlying assumption is that only a subset of variables contributes information to the

model. Keeping uninformative variables in the model adds uncertainty to the prediction. Therefore, it is desirable to exclude them.

The Lasso estimator (Tibshirani, 1996) is a regression estimator where a penalty term is added to the sum of squared residuals in the objective function. This penalty is the L_1 norm of the coefficient estimator, i.e. the sum of the absolute values of the coefficients. Therefore, zero coefficients are favoured for variables which do not contribute relevant information to explain the response. The influence of the penalty term is controlled by a multiplicative tuning constant. The Lasso estimator is widely used. Detailed discussion on recent developments are presented in Hastie et al. (2015).

The penalty in elastic net regression (Zou and Hastie, 2005) is a combination of the penalties of Ridge and Lasso estimators. The property of Lasso to favour zeros in the coefficients is preserved and by the L_2 term of the penalty highly correlated variables tend to obtain similar coefficient estimates. For the elastic net estimator two tuning parameters need to be determined controling the overall influence of the combined penalty and the mixing proportion between L_1 and L_2 penalty.

Even though PLS can be applied to high-dimensional data, the model precision still suffers from uninformative variables. Sparse PLS (SPLS) was introduced by Chun and Keleş (2010) to overcome this problem. The latent variables are constructed as a linear combination of only a subset of the original variables. This method can be applied for regression problems as well as for classification problems.

Sparse optimal scoring (Clemmensen et al., 2012) is another approach to obtain a sparse classification model. A scoring vector transforms the categorical class-membership into a continuous value, which is iteratively optimized. In this framework new developments in regression analysis can be transferred to classification problems.

1.3 Robust modeling for high-dimensional data

Identifying outliers in high-dimensional data can be a challenging task (Filzmoser et al., 2008). So far, only few robust methods exist to model high-dimensional data and to identify model-based outliers, which do not follow the trend of the majority.

The first sparse and robust regression method was introduced by Alfons et al. (2013) and is based on the idea of LTS regression. Only a subset of observations are used for the estimation of the model and sparsity is induced by an L_1 penalty. After a robust estimation is obtained, a reweighting step improves the efficiency of the method. The robust PLS estimator called partial robust M regression (PRM) was introduced

by Serneels et al. (2005). Both the dimension reduction and the regression on latent variables are performed robustly. A sparse MM estimator was introduced in Öllerer (2015) alongside with other robust methods for high-dimensional data.

1.4 Outline of the thesis

The thesis introduces new robust and sparse methods for regression and classification problems. In Chapter 2 a sparse and robust PLS method is introduced. It combines concepts from PRM (Serneels et al., 2005) and SPLS (Chun and Keleş, 2010). A robust subspace is identified, including only a subset of the original variables, where a robust regression model is estimated. This approach is extended to binary classification problems in Chapter 3. Outliers are identified separately for each group. For the robust analysis of multigroup classification problems for high-dimensional data a robust and sparse classifier based on the optimal scoring approach is introduced in Chapter 4. Chapter 5 presents a sparse regression method based on LTS regression with an elastic net penalty. The method is applicable for linear and logistic regression.

The chapters consists of the following publications and submitted papers.

I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust M regression. *Chemometrics and Intelligent Laboratory Systems* 2015; **149**: 50–59.

I. Hoffmann, P. Filzmoser, S. Serneels, K. Varmuza, Sparse and robust PLS for binary classification. *J. Chemometrics* 2016; **30**: 153–162.

I. Hoffmann, P. Filzmoser, C. Croux, Robust and sparse multigroup classification by the optimal scoring approach. Submitted for publication.

F. S. Kurnaz, I. Hoffmann, P. Filzmoser, Robust and sparse estimation methods for high dimensional linear and logistic regression. Submitted for publication.

CHAPTER 2

Sparse partial robust M regression

Abstract: Sparse partial robust M regression is introduced as a new regression method. It is the first dimension reduction and regression algorithm that yields estimates with a partial least squares like interpretability that are sparse and robust with respect to both vertical outliers and leverage points. A simulation study underpins these claims. Real data examples illustrate the validity of the approach.

Key words: Biplot, Partial least squares, Robustness, Sparse estimation

2.1 Introduction

Sparse regression methods have been a major topic of research in statistics over the last decade. They estimate a linear relationship between a predictand $\boldsymbol{y} \in \mathbb{R}^n$ and a predictor data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. Assuming the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.1}$$

the classical estimator is given by solving the least squares criterion

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \tag{2.2}$$

with the squared L_2 norm $\|\boldsymbol{u}\|^2 = \sum_{i=1}^{p} u_i^2$ for any vector $\boldsymbol{u} \in \mathbb{R}^p$. Thereby the predicted responses are $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$. When the predictor data contain a column of ones, the model incorporates an intercept.

Typically, but not exclusively, when p is large, the $\boldsymbol{X}$ data matrix tends to contain columns of *uninformative* variables, i.e. variables that bear no information related to the predictand. Estimates of $\boldsymbol{\beta}$ often have a subset of components $\left\{\hat{\beta}_{j_1}, ..., \hat{\beta}_{j_{\check{p}}}\right\}$ of small magnitude corresponding to $\check{p}$ uninformative variables. As these components are small but not exactly zero, each of them still contributes to the model and, more importantly, to increased estimation and prediction uncertainty. In contrast, a sparse estimator of $\boldsymbol{\beta}$ will have many components that are exactly equal to zero.

Penalized regression methods impose conditions on the norm of the coefficient vector. The Lasso estimate (Tibshirani, 1996), where an L_1 penalty term is used, leads to a sparse coefficient vector:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1, \tag{2.3}$$

with $\|\boldsymbol{u}\|_1 = \sum_{i=1}^{p} |u_i|$ for any vector $\boldsymbol{u} \in \mathbb{R}^p$. The nonnegative tuning parameter λ_1 determines the sparsity of the estimation and implicitly reflects the size of $\check{p}$. The Lasso sparse regression estimate has become a statistical regression tool of widespread application, especially in fields of research where data dimensionality is typically high, such as chemometrics, cheminformatics or bioinformatics (Tibshirani, 2011). But, since it is nonrobust, it may be severely distorted by outliers in the data.

Robust multiple regression has attracted widespread attention from statisticians since as early as the 1970s. For an overview of robust regression methods, we refer to e.g. Maronna et al. (2006). However, only recently, robust sparse regression estimators have been proposed. One of the few existing sparse and robust regression estimators that is robust to both vertical outliers (outliers in the predictand) and leverage points (outliers in the predictor data), is sparse least trimmed squares regression (Alfons et al., 2013), which is a sparse penalized version of the least trimmed squares (LTS) robust regression estimator (Rousseeuw and Leroy, 2003).

In applied sciences, there is often a need for both regression analysis, and interpretative analysis. In order to visualize the data and to interpret the high-dimensional structure(s) in them, it is customary to project the predictor data onto a limited set of latent components and then analyze the individual cases' position as well as how each original variable contributes to the latent components in a biplot. A first approach would be to do a (potentially sparse) principal component analysis, followed by a (po-

tentially sparse) regression. The main issue with that approach is that the principal components are defined according to a maximization criterion that does not account for the predictand. With this reason, partial least squares regression (PLS) (Wold, 1965) has become a mainstay tool in applied sciences such as chemometrics. It provides a projection onto a few latent components that can be visualized in biplots, and it yields a vector of regression coefficients based on those latent components.

Partial least squares regression is both a nonrobust and a nonsparse estimator. Manifold proposals to robustify PLS have been discussed, of which a good overview is given in Filzmoser et al. (2009). One of the most widely applied robust alternatives to PLS is partial robust M regression (Serneels et al., 2005). Likely its popularity is due to the fact that it provides a fair tradeoff between statistical robustness with respect to both vertical outliers and leverage points on the one hand and statistical and computational efficiency on the other hand. From an application perspective, it has been reported to perform well (Liebmann et al., 2010). Introduction of sparseness into the partial least squares framework is a more recent topic of research (Lê Cao et al., 2008; Chun and Keleş, 2010; Allen et al., 2013).

In this article, a novel estimator is introduced, called *Sparse Partial Robust M regression*, which is up to our knowledge the first estimator to offer all three benefits simultaneously: (i) it is based on projection onto latent structures and thereby yields PLS like visualization, (ii) it is integrally sparse, yielding not only regression coefficients with exact zero components, but also sparse direction vectors, and (iii) it is robust with respect to both vertical outliers and leverage points.

2.2 The sparse partial robust M regression estimator

The sparse partial robust M regression (SPRM) estimator can be viewed at as either a sparse version of the partial robust M regression (PRM) estimator (Serneels et al., 2005), or as a way to robustify the sparse PLS (SPLS) estimator (Chun and Keleş, 2010). Therefore, its construction inherits some characteristics from both precursors.

In partial least squares, the latent components (or *scores*) $\boldsymbol{T}$ are defined as linear combinations of the original variables $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{A}$, wherein the so-called *direction vectors* $\boldsymbol{a}_h$ (in the PLS literature also known as *weighting vectors*) are the columns of $\boldsymbol{A}$. The direction vectors maximize squared covariance to the predictand:

$$\boldsymbol{a}_h = \underset{\boldsymbol{a}}{\operatorname{argmax}} \operatorname{cov}^2 (\boldsymbol{X}\boldsymbol{a}, \boldsymbol{y}) , \tag{2.4a}$$

for $h \in \{1, ..., h_{max}\}$ under the constraints that

$$\| \boldsymbol{a}_h \| = 1 \quad \text{and} \quad \boldsymbol{a}_h^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{a}_i = 0 \text{ for } 1 \leq i < h. \tag{2.4b}$$

Here, h_{max} is the maximum number of components we want to retrieve. We assume throughout the article, that both predictor and predictand variables are centered, so that

$$\text{cov}^2 (\boldsymbol{X}\boldsymbol{a}, \boldsymbol{y}) = \frac{1}{(n-1)^2} \boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{y} \boldsymbol{y}^T \boldsymbol{X} \boldsymbol{a} = \frac{1}{(n-1)^2} \boldsymbol{a}^T \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{a}, \tag{2.5}$$

with $\boldsymbol{M} = \boldsymbol{y}^T \boldsymbol{X}$. Regressing the dependent variable onto the scores, yields

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\text{argmin}} \| \boldsymbol{y} - \boldsymbol{T}\boldsymbol{\gamma} \|^2 = \left(\boldsymbol{T}^T \boldsymbol{T}\right)^{-1} \boldsymbol{T}^T \boldsymbol{y}. \tag{2.6}$$

Then, since $\hat{\boldsymbol{y}} = \boldsymbol{T}\hat{\boldsymbol{\gamma}}$ and $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{A}$, one gets $\hat{\boldsymbol{\beta}} = \boldsymbol{A}\hat{\boldsymbol{\gamma}}$.

In order to obtain a robust version of the partial least squares estimator, case weights ω_i are assigned to the rows of $\boldsymbol{X}$ and $\boldsymbol{y}$. Let

$$\tilde{\boldsymbol{X}} = \boldsymbol{\Omega}\boldsymbol{X} \quad \text{and} \quad \tilde{\boldsymbol{y}} = \boldsymbol{\Omega}\boldsymbol{y}, \tag{2.7}$$

with $\boldsymbol{\Omega}$ a diagonal matrix with diagonal elements $\omega_i \in [0, 1]$ for $i \in \{1, ..., n\}$. Outlying observations will receive a weight lower than one. An observation is an outlier when it has a large residual, or a large value of the covariate (hence a large leverage) in the latent regression model (i.e. the regression of the predictand on the latent components). Let $\boldsymbol{t}_i$ denote the rows of $\boldsymbol{T}$, $r_i = y_i - \boldsymbol{t}_i^T \hat{\boldsymbol{\gamma}}$ are the residuals of the latent variable regression model, where y_i are the elements of the vector $\boldsymbol{y}$. Let $\hat{\sigma}$ denote a robust scale estimator of the residuals; we take the median absolute deviation (MAD). Then the weights are defined by

$$\omega_i^2 = \omega_R \left(\frac{r_i}{\hat{\sigma}}\right) \omega_T \left(\frac{\|\boldsymbol{t}_i - \text{med}_j(\boldsymbol{t_j})\|}{\text{med}_i \|\boldsymbol{t}_i - \text{med}_j(\boldsymbol{t_j})\|}\right). \tag{2.8}$$

More specifics on weight functions ω_R and ω_T will be discussed in Section 2.3.

With (2.5) and $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{y}}^T \tilde{\boldsymbol{X}}$, the robust maximization criterion for the direction vectors is

$$\boldsymbol{a}_h = \underset{\boldsymbol{a}}{\text{argmax}}\, \boldsymbol{a}^T \tilde{\boldsymbol{M}}^T \tilde{\boldsymbol{M}} \boldsymbol{a}, \tag{2.9a}$$

under the constraints that

$$\| \boldsymbol{a}_h \| = 1 \quad \text{and} \quad \boldsymbol{a}_h^T \tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}} \boldsymbol{a}_i = 0 \text{ for } 1 \leq i < h, \tag{2.9b}$$

which is identical to maximization criterion (2.4) if $\boldsymbol{\Omega}$ is the identity matrix.

In order to obtain a fully robust PLS estimation, the latent variable regression needs to be robustified too. Thereunto, note that the ordinary least squares minimization criterion can be written as

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(y_i - \boldsymbol{t}_i^T \boldsymbol{\gamma}\right), \tag{2.10}$$

with $\rho(u) = u^2$. Using a ρ function with bounded derivative in criterion (2.10) yields a well-known class of robust regression estimators called *M estimators*. They are computed as iteratively reweighted LS-estimators, with weight function $\omega(u) = \rho'(u)/u$. The resulting estimator is the partial robust M regression estimator (Serneels et al., 2005).

Imposing sparseness on the PRM estimator can now be achieved by setting an L_1 penalty to the direction vectors $\boldsymbol{a}_h$ in (2.9a). To get sufficiently sparse estimates the sparseness is imposed on a surrogate direction vector $\boldsymbol{c}$ instead (Zou et al., 2006). More specifically,

$$\min_{\boldsymbol{c},\boldsymbol{a}} -\kappa \boldsymbol{a}^T \tilde{\boldsymbol{M}}^T \tilde{\boldsymbol{M}} \boldsymbol{a} + (1-\kappa)(\boldsymbol{c}-\boldsymbol{a})^T \tilde{\boldsymbol{M}}^T \tilde{\boldsymbol{M}} (\boldsymbol{c}-\boldsymbol{a}) + \lambda_1 \parallel \boldsymbol{c} \parallel_1 \tag{2.11a}$$

under the constraints that

$$\parallel \boldsymbol{a}_h \parallel = 1 \quad \text{and} \quad \boldsymbol{a}_h^T \tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}} \boldsymbol{a}_i = 0 \text{ for } 1 \leq i < h. \tag{2.11b}$$

The final estimate of the direction vector is given by

$$\boldsymbol{a}_h = \frac{\hat{\boldsymbol{c}}}{\|\hat{\boldsymbol{c}}\|}, \tag{2.12}$$

with $\hat{\boldsymbol{c}}$ is the surrogate vector minimizing (2.11a). In this way, we obtain a sparse matrix of robustly estimated direction vectors $\boldsymbol{A}$ and scores $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{A}$. After regressing the dependent variable on the latter using criterion (2.10) we get the sparse partial robust M regression estimator. Note that the sparsity of the estimated directions carries through to the vector of regression coefficients.

This definition leads to a complex optimization task in which three parameters need to be selected h_{max}, κ and λ_1. Nevertheless, Chun and Keleş (2010) have shown that the optimization problem does not depend on κ for any $\kappa \in (0, 1/2]$ for univariate $\boldsymbol{y}$ (which is the case throughout this article). Therefore, the three parameter search reduces to the number of latent components h_{max} and the sparsity parameter λ_1. How these parameters can be selected will be discussed in detail in Section 2.4. The next section outlines a fast algorithm to compute the SPRM estimator.

2.3 The SPRM algorithm

The SPRM estimator can be implemented in a surprisingly straightforward manner. Chun and Keleş (2010) have shown that imposing sparsity on PLS estimates according to criterion (2.11) yields analytically exact solutions. Denote by $\boldsymbol{z}_h$ the classical, nonsparse PLS direction vectors of the deflated $\boldsymbol{X}$ matrix, i.e. $\boldsymbol{z}_h = \boldsymbol{E}_h^T\boldsymbol{y} / \parallel \boldsymbol{E}_h^T\boldsymbol{y} \parallel$, wherein $\boldsymbol{E}_h$ is $\boldsymbol{X}$ deflated in order to fulfil the orthogonality side constraints in (2.11b). Hence, $\boldsymbol{E}_1 = \boldsymbol{X}$ and $\boldsymbol{E}_{h+1} = \boldsymbol{E}_h - \boldsymbol{t}^h\boldsymbol{t}^{hT}\boldsymbol{E}_h/\|\boldsymbol{t}^h\|^2$ where $\boldsymbol{t}^h$ is the score vector computed in the previous step. Then the exact SPLS solution is given by

$$\boldsymbol{w}_h = (|\boldsymbol{z}_h| - \lambda_1/2) \odot \boldsymbol{I}\left(|\boldsymbol{z}_h| - \lambda_1/2 > 0\right) \odot \operatorname{sgn}(\boldsymbol{z}_h), \tag{2.13}$$

wherein $\boldsymbol{I}(\cdot)$ denotes the indicator function that yields a vector whose elements equal 1 if the argument is true and 0 otherwise, and $\odot$ denotes the Hadamard (element wise) vector product. In (2.13), $|\boldsymbol{z}_h|$ is the vector of the absolute values of the components of $\boldsymbol{z}_h$, and $\operatorname{sgn}(\boldsymbol{z}_h)$ is the vector of the signs of the components. By putting the vectors $\boldsymbol{w}_h$ in the columns of $\boldsymbol{W}$ for $h = 1, ..., h_{max}$, the sparse direction vectors in terms of the original, not deflated variables are given by $\boldsymbol{A} = \boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{W})^{-1}$.

Formula (2.13) can be replaced by an equivalent expression. Let η denote a tuning parameter with $\eta \in [0, 1)$. Then we redefine

$$\boldsymbol{w}_h = \left(|\boldsymbol{z}_h| - \eta \max_i |z_{ih}|\right) \odot \boldsymbol{I}\left(|\boldsymbol{z}_h| - \eta \max_i |z_{ih}| > 0\right) \odot \operatorname{sgn}(\boldsymbol{z}_h), \tag{2.14}$$

with z_{ih} being the components of $\boldsymbol{z}_h$. The parameter η determines the size of the threshold, as a fraction of the maximum of $\boldsymbol{z}_h$, beneath which all elements of vector $\boldsymbol{w}_h$ are set to zero. Since the range of η is known in this definition, it facilitates the tuning parameter selection via cross validation (see Section 2.4).

Computation of the M estimators in (2.10) boils down to iteratively reweighting the least squares estimator. We use the redescending Hampel weighting function giving a good trade-off between robustness and efficiency (Hampel et al., 1986).

$$\omega(x) = \begin{cases} 1 & |x| \le a \\ \frac{a}{|x|} & a < |x| \le b \\ \frac{q-x}{q-b}\frac{a}{|x|} & \text{if} \quad b < |x| \le q \\ 0 & q < |x| \end{cases}, \tag{2.15}$$

wherein the tuning constants a, b and q can be chosen as distribution quantiles. For the residual weight function ω_R in (2.8) we take the $0.95, 0.975$ and 0.999 quantiles of the standard normal, for ω_T the corresponding quantile of a chi-square distribution.

Algorithm 2.1: The SPRM algorithm.

$\boldsymbol{X}$ and $\boldsymbol{y}$ denote robustly centered data (by column-wise median).

1. Calculate initial case weights:
   - Calculate distances for $\boldsymbol{x}_i$ (ith row of $\boldsymbol{X}$) and y_i:
   $$d_i = \frac{\|\boldsymbol{x}_i\|}{\text{med}_j \|\boldsymbol{x}_j\|} \quad \text{and}$$
   $$r_i = \frac{|y_i|}{c\,\text{med}_j |y_j|} \quad \text{for } i \in \{1, ..., n\}$$
   where $c = 1.4826$ for consistency of the MAD.
   - Define initial weights $\omega_i = \sqrt{\omega_T(d_i)\omega_R(r_i)}$ for $\boldsymbol{\Omega}$ (see (2.8)).
2. Iteratively reweighting:
   - Weight data:
   $$\begin{aligned} \boldsymbol{X}_\omega &= \boldsymbol{\Omega}\boldsymbol{X} \\ \boldsymbol{y}_\omega &= \boldsymbol{\Omega}\boldsymbol{y} \end{aligned}$$
   - Apply the sparse NIPALS to $\boldsymbol{X}_\omega$ and $\boldsymbol{y}_\omega$ and obtain scores $\boldsymbol{T}_\omega$, directions $\boldsymbol{A}_\omega$, coefficients $\hat{\boldsymbol{\beta}}_\omega$ and predicted response $\hat{\boldsymbol{y}}_\omega$.
   - Calculate weights for scores and response.
     - Center $\text{diag}(1/\omega_1, ..., 1/\omega_n)\boldsymbol{T}_\omega$ by the median and scale the columns with the Rousseeuw and Croux (1993) robust scale estimator Qn to obtain $\tilde{\boldsymbol{T}}$.
     - Calculate distances for $\tilde{\boldsymbol{t}}_i$ (ith row of $\tilde{\boldsymbol{T}}$) and the robustly centered and scaled residuals r_i for $i \in \{1, ..., n\}$:
     $$d_i = \frac{\|\tilde{\boldsymbol{t}}_i\|}{\text{med}_j \|\tilde{\boldsymbol{t}}_j\|}$$
     $$r_i = \frac{|y_{\omega,i} - \hat{y}_{\omega,i} - \text{med}_k(y_{\omega,k} - \hat{y}_{\omega,k})|}{c\,\text{med}_j |y_{\omega,j} - \hat{y}_{\omega,j} - \text{med}_k(y_{\omega,k} - \hat{y}_{\omega,k})|}$$
     - Update weights $\omega_i = \sqrt{\omega_T(d_i)\omega_R(r_i)}$.

   Repeat until convergence of $\hat{\boldsymbol{\beta}}_\omega$.
3. Denote estimates of the final iteration by $\boldsymbol{A}$ and $\hat{\boldsymbol{\beta}}$ and the scores by $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{A}$.

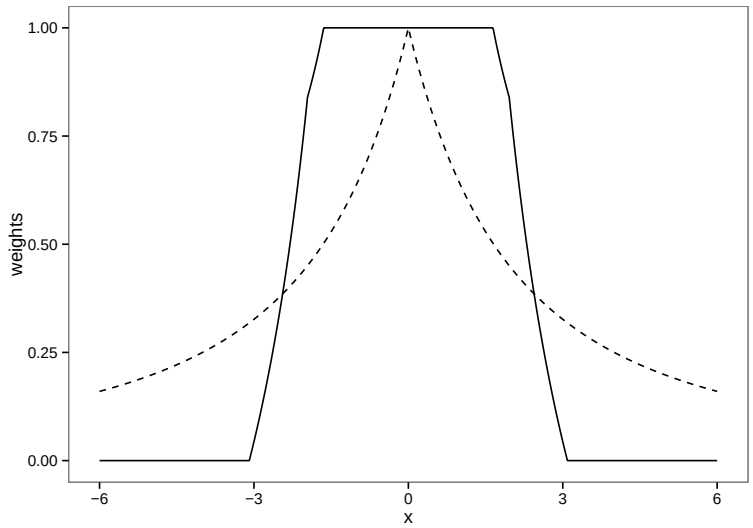


Figure 2.1: The Hampel (solid) weighting function with standard normal 95%, 97.5% and 99.9% quantiles as cutoffs and the Fair (dashed) weighting function with parameter $c = 4$.

Note that in the original publication on partial robust M regression (Serneels et al., 2005), the Fair function was recommended (both weighting functions are plotted in Figure 2.1), but the authors consider the Hampel redescending function superior over the Fair function, because (i) it yields case weights that are much easier to interpret, since they are exactly 1 for the regular cases, exactly 0 for the severe outliers and in the interval (0,1) for the moderate outliers and because (ii) the tuning constants for the cutoffs can be set according to intuitively understandable statistical values such as quantiles from a corresponding distribution function.

The algorithm to compute the SPRM estimators iteratively reweights a sparse PLS estimate. This sparse PLS estimate is computed as in Lee et al. (2011), who outline a sparse adaptation of the NIPALS computation scheme (Wold, 1975), where in each step of the NIPALS the obtained direction vector of the deflated $\boldsymbol{X}$ matrix is modified according to Equation (2.14) in order to get sparseness. The starting values of the SPRM algorithm have to be robust. Failing to estimate robust starting values, would lead to an overall nonrobust estimator. Algorithm 2.1 presents the computing scheme and details the starting values. We iterate until convergence, that is whenever the *relative* difference in norm between two consecutive approximations of $\hat{\boldsymbol{\beta}}$ is smaller than a specified threshold, e.g. 10^{-2}. An implementation of the algorithm is available on

CRAN in the package sprm (Serneels and Hoffmann, 2014).

2.4 Model selection

The computation of the SPRM estimator requires specification of h_{max}, the number of latent components, and the sparsity parameter $\eta \in [0, 1)$ (see Equation (2.14)). For $\eta = 0$ the model is estimated including all variables, for η tending towards 1 almost no variables are selected.

A grid of values for η is searched and $h_{max} = 1, 2, ..., H$. With k-fold robust cross validation the best parameter combination is selected. For each combination of h_{max} and η the model is estimated k times based on a training set containing $(100-k)$ percent of the data, and then evaluated for the remaining data, constituting the validation set. All observations are considered once for validation and so we obtain a single prediction for each of them. As robust cross validation criterion the one sided $\alpha\%$ trimmed mean is calculated from the squared prediction errors, such that the largest $\alpha\%$ errors which may come from outliers, are excluded. We choose the parameter combination where this measure of prediction accuracy is smallest.

The model selection procedure in the following is based on 10-fold cross validation. For the robust methods, the one sided 15% trimmed mean squared error is applied as decision criterion and for the classical methods the mean squared error of prediction is used for validation. The parameter h_{max} has a value domain from 1 to 5 and for SPLS and SPRM the sparsity parameter η is chosen among ten equally spaced values from 0 to 0.9.

2.5 Simulation study

In this section the properties of SPRM and the related methods PRM, PLS and SPLS are studied by means of a simulation study. The predictand is generated according to the model

$$y_i = \boldsymbol{t}_i \boldsymbol{\gamma} + e_i \quad \text{for} \quad 1 \leq i \leq n, \tag{2.16}$$

where the score matrix $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{A}$, for a given matrix of direction vectors $\boldsymbol{A}$.

Let $\boldsymbol{X}$ be an $n \times p$ data matrix with columns generated independently from the standard normal distribution. We generate the columns $\boldsymbol{a}_h$ $(h = 1, \ldots, h_{max})$ of $\boldsymbol{A}$ such that only the first $q \leq p$ elements of each $\boldsymbol{a}_h$ are nonzero. Thereby, the data matrix $\boldsymbol{X}$ is divided into q columns of relevant variables and $p - q$ columns of uninformative

variables. The nonzero part of $\boldsymbol{A}$ is given by the eigenvectors of the matrix $\boldsymbol{X}_q^T\boldsymbol{X}_q$, where $\boldsymbol{X}_q$ contains the first q columns of $\boldsymbol{X}$. This ensures that the side conditions for $\boldsymbol{a}_h$ hold (see (2.11b)). The components of the regression vector $\boldsymbol{\gamma} \in \mathbb{R}^{h_{max}}$ are drawn from the uniform distribution on the interval $[0.5, 1.5]$. The errors e_i are generated as independent values from the standard normal distribution. In a second experiment, we investigate the influence of outliers. The first 10% of the errors are generated from $N(15, 1)$ instead of $N(0, 1)$. To induce bad leverage points, the first 5% of the observations $\boldsymbol{x}_i$ are replaced by vectors of random values from $N(5, 0.1)$. This will demonstrate the stability of the robust methods when compared to the classical approaches.

In the simulation study, $m_{rep} = 200$ data sets with $n = 60$ observations are generated according to (2.16) for various values of p. While $q = 6$ is fixed, we will increase p gradually and therefore decrease the signal to noise ratio. This illustrates the effect of uninformative variables on the four model estimation methods and incorporates low dimensional as well as high-dimensional settings. For every generated data set we compute the estimator $\hat{\boldsymbol{\beta}}^j$ (for $1 \leq j \leq m_{rep}$) with sparsity parameter η and h_{max} selected as described in Section 2.4. Note that the true coefficients $\boldsymbol{\beta}^j$ are different for every simulation run, since every data set is generated with a different regression vector $\boldsymbol{\gamma}$.

Performance Measures: To evaluate the simulation results, the mean squared error (MSE) is used as a measure of the accuracy of the model estimation.

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}) = \frac{1}{m_{rep}} \sum_{1 \leq j \leq m_{rep}} \|\hat{\boldsymbol{\beta}}^j - \boldsymbol{\beta}^j\|^2 \tag{2.17}$$

Furthermore, let $\hat{\boldsymbol{\beta}}_0^j$ be the subvector of $\boldsymbol{\beta}^j$ corresponding to the uninformative variables. In the true model $\boldsymbol{\beta}_0^j$ is a vector of zeros. Nonzero values of $\hat{\boldsymbol{\beta}}_0^j$ contribute to the model uncertainty. One main advantage of sparse estimation is to reduce this uncertainty by setting most coefficients of uninformative variables exactly to zero. The mean number of nonzero values in $\hat{\boldsymbol{\beta}}_0^j$ is reported for both sparse methods to illustrate whether this goal was achieved. Furthermore, the mean number of nonzero coefficients of the informative variables, is reported.

The last quality criterion discussed in this section is the prediction performance of the estimated model for new data of the same structure. A test data set with $n = 60$ observations is generated according to the model in each repetition. For $1 \leq j \leq m_{rep}$ the estimated response of the test data is denoted by $\hat{\boldsymbol{y}}_{test}^j$ and the true response is $\boldsymbol{y}_{test}^j$.

Table 2.1: Mean percentage of correct zero coefficients, i.e. zero coefficients of uninformative variables, for SPLS and SPRM for simulations with (a) clean training data and (b) training data with 10% outliers.

(a) without outliers

$p-q$	20	100	200	300	500
SPLS	91.2	97.6	98.4	99.1	98.0
SPRM	75.5	93.4	95.1	96.8	94.5

(b) with outliers

$p-q$	20	100	200	300	500
SPLS	43.5	38.7	36.2	39.3	35.5
SPRM	76.9	91.0	94.2	98.1	97.6

Table 2.2: Mean percentage of correct nonzero coefficients, i.e. nonzero coefficients of the six informative variables, for SPLS and SPRM for simulations with (a) clean training data and (b) training data with 10% outliers.

(a) without outliers

$p-q$	20	100	200	300	500
SPLS	65.8	54.1	52.0	48.4	46.6
SPRM	70.0	53.8	47.8	46.7	44.8

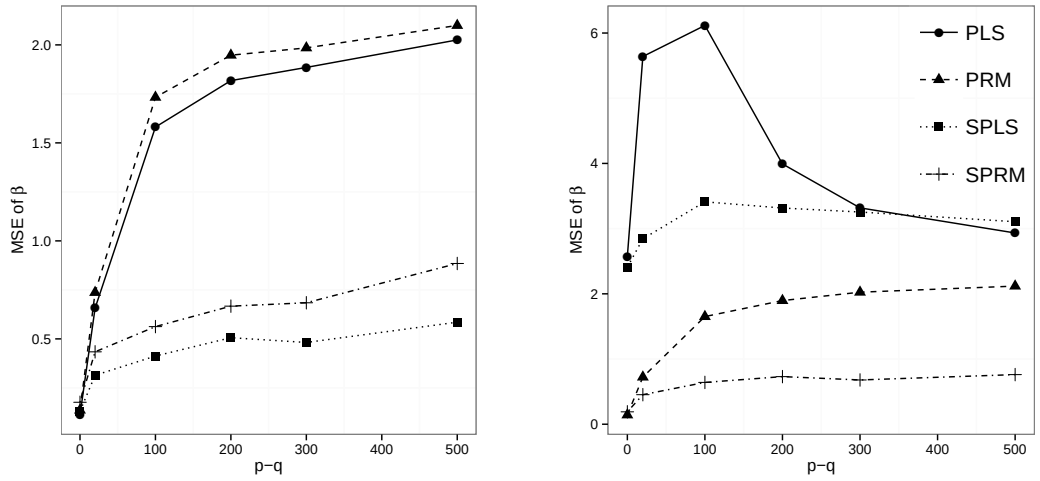
(b) with outliers

$p-q$	20	100	200	300	500
SPLS	65.2	70.1	71.3	68.4	72.2
SPRM	68.8	53.8	49.3	45.0	41.3

Then the mean squared prediction error (MSPE) is computed as

$$\text{MSPE} = \frac{1}{m_{rep}} \sum_{1 \le j \le m_{rep}} \|\hat{\boldsymbol{y}}^j_{test} - \boldsymbol{y}^j_{test}\|^2. \tag{2.18}$$

Results for clean data: In the absence of outliers (see Figure 2.2a and 2.3a), the overall performance of the classical methods SPLS and PLS is slightly better than for the robust counterparts SPRM and PRM, respectively. In Figure 2.2a it is seen that the MSE is smallest for SPLS. If all variables are informative, so $p-q=0$, then PLS performs as good as SPLS; but for an increasing number of uninformative variables PLS quickly becomes less reliable. The same can be observed for the mean squared prediction error in Figure 2.3a. Both Figures 2.2a and 2.3a show that SPRM is not as accurate



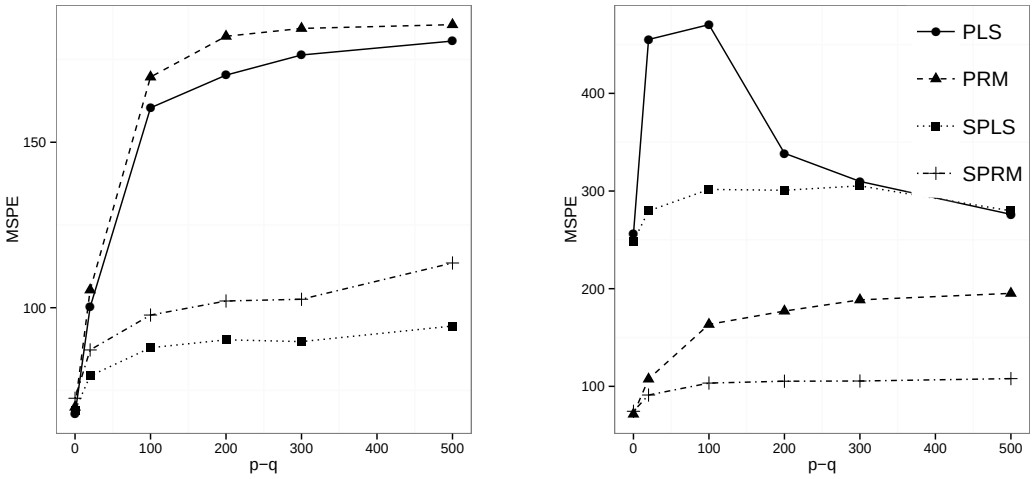
(a) without outliers (b) with outliers

Figure 2.2: Mean squared error of the coefficient estimates for PLS, PRM, SPLS and SPRM for simulations with (a) clean training data and (b) training data with 10% outliers.

as SPLS, but performs much better than PLS and PRM for settings with increasing number of uninformative variables.

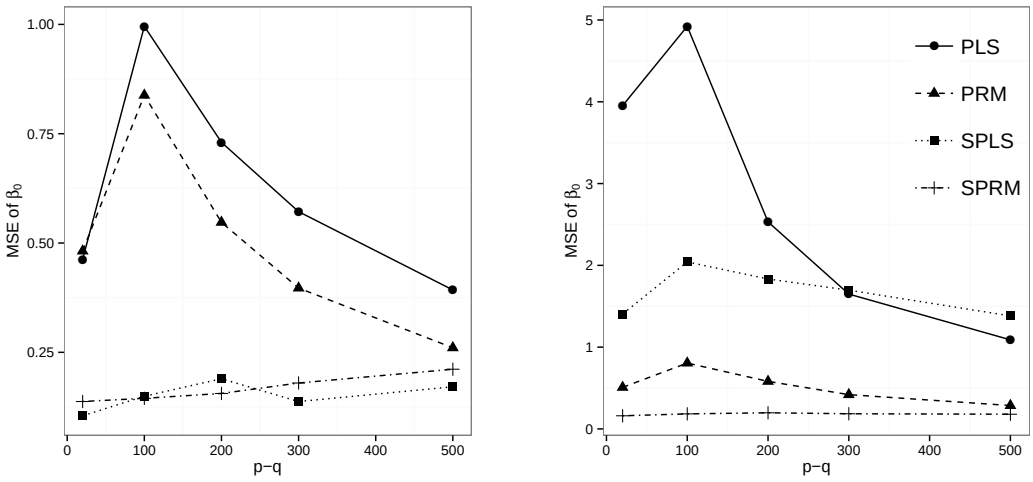
Table 2.1a underpins the advantage of sparse methods. It shows that the average percentage of uninformative variables excluded from the model is close to 100%. SPLS is again slightly better than SPRM, but for both estimates few uninformative variables are included, leading to reduced estimation error in comparison to PLS and PRM. The MSE for the estimation of $\boldsymbol{\beta}_0$ is given in Figure 2.4a. SPLS and SPRM have a comparably good performance, even though SPRM has less zero components in $\hat{\boldsymbol{\beta}}_0^j$. That means that the nonzero coefficient estimates of the uninformative variables are very small for SPRM. PRM gives surprisingly good results for the MSE of $\hat{\boldsymbol{\beta}}_0$ and outperforms PLS. Table 2.2a shows the mean percentage of nonzero coefficients for the informative variables. For both SPLS and SPRM only roughly half of the six informative variables are included. The sensitivity of SPLS, i.e. the proportion of nonzero correctly identified as such, is reported to be close to 100% in other simulation settings (Chun and Keleş, 2010), but in this simulation setting the true nonzero coefficients can be close to zero. SPRM includes slightly less variables, but gives very comparable results to SPLS.

Results for data with outliers: Outliers distort the estimation of PLS and SPLS heavily. Figures 2.2b and 2.3b show that the performance of PLS and SPLS strongly de-



(a) without outliers (b) with outliers

Figure 2.3: Mean squared prediction error for PLS, PRM, SPLS and SPRM for simulations with (a) clean training data and (b) training data with 10% outliers.



(a) without outliers (b) with outliers

Figure 2.4: Mean squared error of the coefficient estimates of the uninformative variables for PLS, PRM, SPLS and SPRM for simulations with (a) clean training data and (b) training data with 10% outliers.

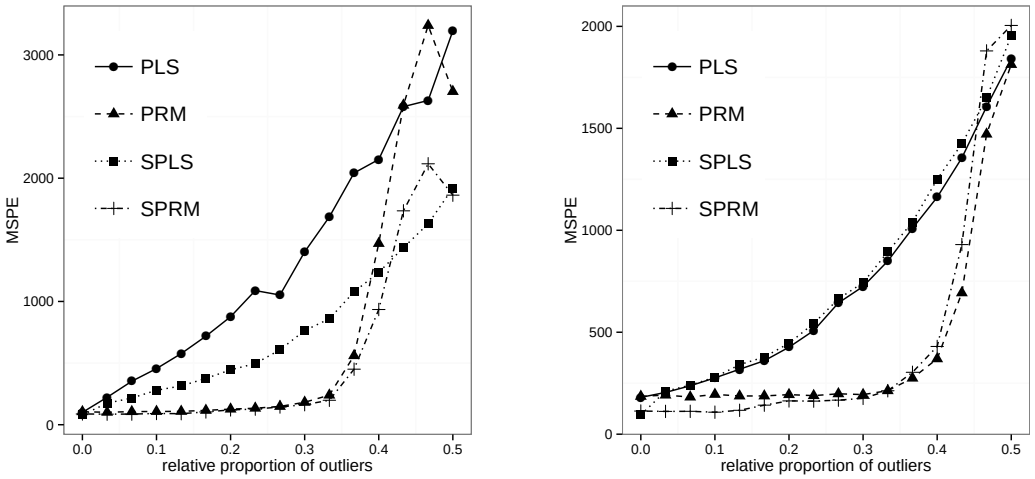
teriorates, while the robust methods are hardly influenced by the presence of the outliers. Furthermore, the robust methods behave as expected as the number of uninformative variables increases: The MSE and MSPE for PRM increase remarkably, whereas SPRM shows only a slight increase, which illustrates the advantage of sparse estimation.

In Table 2.1b it is seen that SPRM excludes nearly all uninformative variables from the model, whereas SPLS fails to identify them up to a high degree. For all settings, less than half of the uninformative variables are excluded. Hence, the estimation of $\boldsymbol{\beta}_0$ is distorted for the classical methods as shown in Figure 2.4b. Not only the uninformative variables are affected by this trend. In Table 2.2b, the average percentage of nonzero coefficients corresponding to informative variables, are shown. From these results, it is evident that SPLS includes more informative variables compared to the case without outliers, which can be explained by the relatively large number of contributing variables in the models. For SPRM only marginal changes are observed compared to the results for data without outliers.

Increasing the number of outliers: An important focus in the analysis of robust methods, is to study how an increasing percentage of outliers affects the model estimation. We use the same simulation design, again with $m_{rep} = 200$ repetitions for each considered number of outliers. In each step the number of outliers increases by two (one of these two is a bad leverage point) till 50% outliers are generated. The mean squared prediction error as defined in (2.18) is calculated. Figures 2.5a and 2.5b display the MSPE for increasing number of outliers, each graph for a fixed number of uninformative variables.

We observe for the robust methods PRM and SPRM hardly any change in the quality of the prediction performance of the estimated models for up to 33% contamination. The classical methods yield distorted results even for only 3% contamination. Figure 2.5b show that this high robustness of PRM and SPRM remains when there is a large number of (uninformative) variables. We conclude that the robust methods clearly outperform PLS and SPLS in presence of outliers, while SPRM gives better mean squared prediction error than PRM for percentages of outliers up to 33 percent.

Nonnormal error distributions: A common assumption in model (2.16) is that the errors e_i come from a normal distribution. We simulate data with $p = 500$, $q = 6$ and $n = 60$ as described previously for the setting without outliers, but replace the error term e_i by random values from heavy tailed distributions. In Table 2.3, the MSE of the coefficients MSE($\hat{\boldsymbol{\beta}}$) estimated for simulated data with normal error terms are



(a) with $p - q = 20$ (b) with $p - q = 500$

Figure 2.5: Mean squared prediction error for PLS, PRM, SPLS and SPRM illustrating the effect of increasing number of outliers for data with (a) 20 uninformative variables, (b) 500 uninformative variables.

Table 2.3: Mean value (and standard error) of the mean squared error of the coefficient estimates $\text{MSE}(\hat{\boldsymbol{\beta}})$ for simulated data with error terms from the standard normal distribution, the t distribution with 3 and 2 degrees of freedom and Laplace distribution with dispersion parameter 1 and 2.

	N		t_3		t_2		L1		L2	
PLS	2.00	(.05)	2.25	(.06)	2.77	(.11)	2.04	(.05)	2.72	(.05)
PRM	2.10	(.05)	2.13	(.05)	2.18	(.05)	2.06	(.05)	2.31	(.06)
SPLS	0.60	(.02)	1.29	(.07)	2.29	(.15)	1.06	(.04)	2.94	(.09)
SPRM	0.88	(.04)	1.16	(.05)	1.22	(.05)	1.15	(.05)	2.43	(.09)

compared to those from data with error terms from the t distribution with 3 and 2 degrees of freedom and the Laplace distribution with a dispersion parameter of 1 and of 2. The mean squared error of the coefficient estimates behaves as expected for the t distributions. It increases significantly for the classical methods as the number of degrees of freedom decreases from three to two, in which case more extreme values are generated. For error terms from the Laplace distribution, the advantage of the robust methods gets more pronounced for the Lapalce distribution with the higher dispersion parameter, which generates more extreme values.

Computation time and convergence: The computation time of the robust sparse NIPALS algorithm strongly depends on the number of iterations needed till convergence

of the robust and sparse coefficient estimates $\hat{\boldsymbol{\beta}}_\omega$ as described in Algorithm 2.1 in step 2. This varies with the structure of the data. In the simulation study with ten percent outliers in the data and $p = 500$ on average five iterations are needed, i.e. the sparse NIPALS algorithm, which is computationally efficient for a univariate response, has to be applied on average five times. On a standard PC (Intel i7-4790K) the average computation time for the estimation of a model based on these data with fixed parameters $\eta = 0.5$ and $h_{max} = 2$ is 0.12 seconds.

2.6 Application

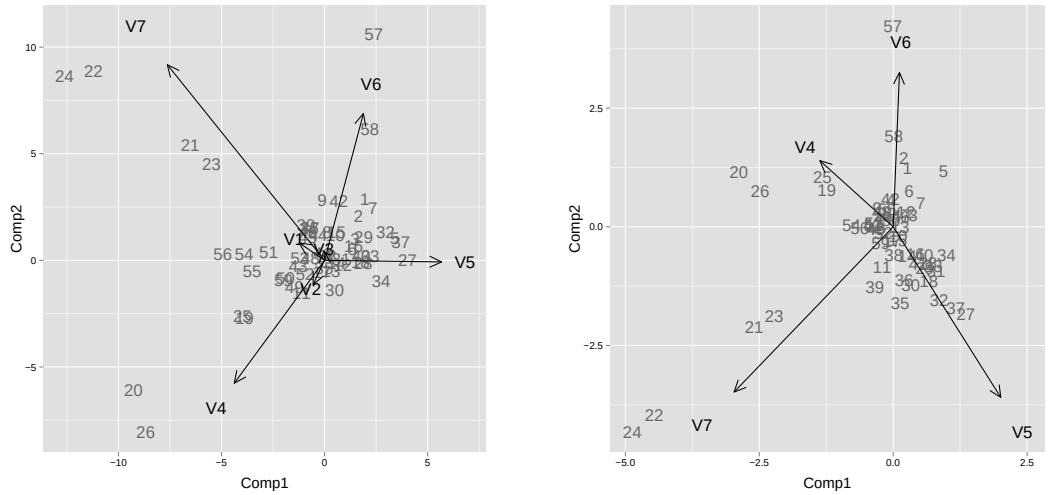
Sparse regression methods and big data go hand in hand. Therefore, there are manifold applications of those methods in the *omics* fields (e.g. the microarray CHIP-chip data (Chun and Keleş, 2010)), but they have also found their way into chemometrics (e.g. Filzmoser et al., 2012) or medicine (e.g. the application on NMR spectra of neural cells (Allen et al., 2013)). Even though sparse regression methods are of great use when data dimensionality is high, they can already be beneficial when applied to low dimensional problems (which, in the context of classification, has been reported in Filzmoser et al. (2012)). Therefore, in the first example we will focus on data of moderate dimensionality, followed by a gene expression example to illustrate the application to high-dimensional data.

The gloss data: The data consist of $n = 58$ polymer stabilization formulations, wherein the $p = 7$ predictors are the respective concentrations of seven different classes of stabilizers. The actual nature of the classes of stabilizers, as well as the respective concentrations, are proprietary to BASF Corp. and cannot be disclosed. The response variable targets to quantify the quality of stabilization by measuring how long it takes for the polymer to lose 50% of its gloss when weathered (in what follows, simply called the *gloss*). The target is to predict the gloss from the stabilizer formulations. The data were scaled with the Qn scale for the robust methods (Rousseeuw and Croux, 1993) and for the classical methods with the standard deviation.

PLS, SPLS, PRM and SPRM use the 10-fold cross validation procedure described in

Table 2.4: Prediction performance for polymer stabilizer data.

	PLS	PRM	SPLS	SPRM
15% TMSPE	2099382	2218181	2113960	2047858



(a) PRM (b) SPRM

Figure 2.6: The PRM and SPRM biplots for the gloss data example.

Section 2.4. The optimal number of latent components for PLS and PRM was detected to equal 1. For SPRM the optimal number of latent components is 4 and the sparsity parameter was found to be $\eta = 0.6$; for SPLS we have $h_{max} = 3$ and $\eta = 0.9$.

To evaluate the four methods, leave-one-out cross validation was performed and the one sided 15% trimmed mean squared prediction error (TMSPE) is reported in Table 2.4. SPRM performs slightly better according to the TMSPE. Another advantage of sparse robust modeling in this example is the interpretability. Figure 2.6 compares the biplots of PRM and SPRM for the first two latent components. In the sparse biplot variables V1, V2 and V3 are excluded and so it is easier to grasp in which way the latent components depend on the original variables, and how the individual cases differ with respect to the selected variables.

The NCI data: The National Cancer Institute provides data sets of measurements from 60 human cancer cell lines (`http://discover.nci.nih.gov/cellminer/`). The 40th observation has to be excluded due to missing values, i.e. $n = 59$. The gene expression data comes from an Affymetrix HG-U133A chip and was normalized with the GCRMA method. It is used to model log_2 transformed protein expression from a Lysate Array. From the gene expression data only the 25% of the variables with highest variance are considered, which leads to $p = 5571$, as was similarly conducted by Lee et al. (2011). The protein data consists of measurements of 162 expression levels. Since the proposed method is designed for univariate response we modeled the relationship for

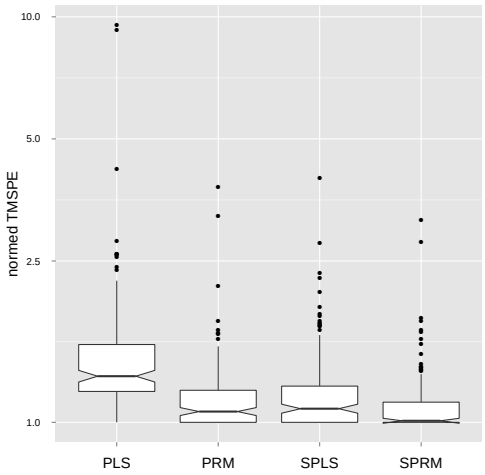


Figure 2.7: Boxplots of normed TMSPE of 162 responses from the NCI data for PLS, PRM, SPLS and SPRM.

each protein expression separately and obtain 162 models for each of the competitive methods.

As before, the model selection is done using 10-fold cross validation (see Section 2.4) and the selected models are evaluated with the 15% TMSPE. For each of the 162 different responses the TMSPE of each estimated model is divided by the smallest of the four TMSPEs. This normed TMSPE is a value equal to 1 (for the best method) or larger and we can compare it across the different responses (see Figure 2.7). Overall, the combination of sparsity and robustness leads to a superior evaluation. The median of the normed TMSPE of the SPRM models is very close to 1 and therefore, we can conclude that for half of the models SPRM is either the best or very close to the best model. PLS is not an appropriate method for these data, since the TMSPE differs strongly from the best model in most cases.

For purpose of illustration, we focus on Keratin 18 as response. It has the highest variance of all responses and its expression is an often used criterion for the detection of carcinomas (Oshima et al., 1996). Table 2.5 presents the number of latent components and the number of selected variables (i.e. having nonzero estimated coefficients) for each method, together with the TMSPE. The SPRM model gives the best result with only 6 out of 5571 variables selected. Even PRM performs better than SPLS in this high-dimensional setting, which underpins the importance of robust estimation for these data. Figure 2.8 shows the biplot of scores and directions for the first two latent components

Table 2.5: Model properties for NCI gene expression data with protein expression of Keratin 18 as response variable.

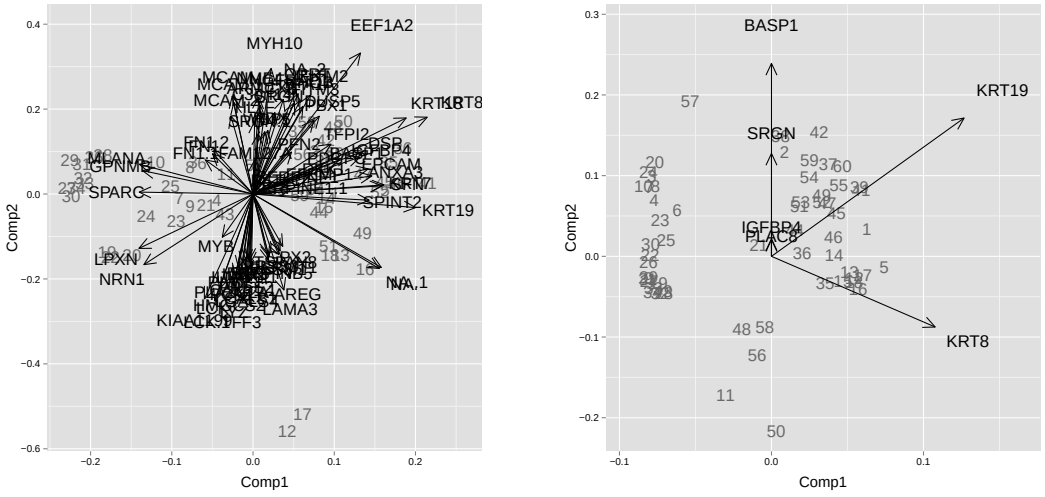
	PLS	PRM	SPLS	SPRM
15% TMSPE	3.22	1.72	2.03	1.24
no. of latent components	4	2	2	3
no. of selected variables	5571	5571	78	6

of the SPLS and the SPRM model. For SPRM, the first latent component is determined by the variables KRT8 and KRT19. The expression of these genes is known to be closely related to the protein expression of Keratin 18 and they are used for the identification and classification of tumor cells (Schelfhout et al., 1989; Oshima et al., 1996). KRT8 has previously been reported to play an important role in sparse and robust regression models of these data (Alfons et al., 2013). The biplot further unveils some clustering in the scores and provides insight into the multivariate structure of the data. The biplot of the SPLS model (Figure 2.8a) cannot be interpreted since this model including 78 variables is too complex. Interestingly, in the SPLS biplot KRT8 and KRT19 are also the genes which have the largest positive influence on the first latent component.

Note that the case weights ω_i of the robust models presented in Figure 2.9 are as expected: they are one for the bulk of the data, exactly zero for the potential outliers and in the interval (0,1) for a few observations, which is an immediate consequence of adopting the Hampel weighting function (Equation (2.15) and Figure 2.1). Hence, outliers can easily be identified. The detection of potential outliers differs between PRM and SPRM, but the pattern is similar.

2.7 Conclusions

SPRM is a sparse and robust regression method, which performs dimension reduction in a manner closely related to partial least squares regression. It performs intrinsic variable selection and retrieves sparse latent components, which can be visualized in biplots and interpreted better than nonsparse latent components especially for high-dimensional data. Since sparse methods eliminate the uninformative variables, higher estimation and prediction accuracy is attained. The SPRM estimation of latent components and the selection of variables is resistant to outliers. To reduce the influence of outliers on the model estimation, an iteratively reweighted regression algorithm is used. The resulting case weights can be used for outlier diagnostics.



(a) SPLS (b) SPRM

Figure 2.8: The SPLS and SPRM biplots for the gene data example with protein expression of Keratin 18 as response.

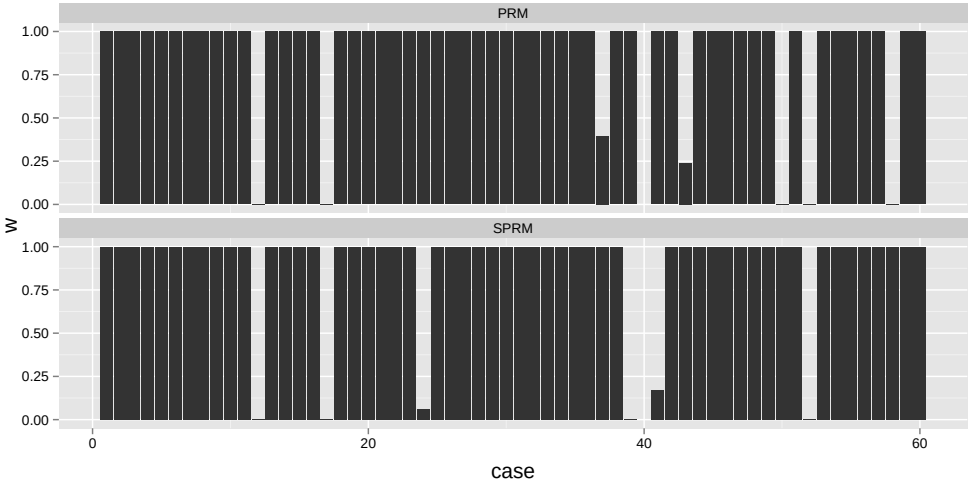


Figure 2.9: The PRM and SPRM case weights for the gene data example with protein expression of Keratin 18 as response.

We demonstrated the importance of robustness and sparsity properties in a simulation study. The method was shown to be robust with respect to outliers in the predictors and in the response and achieved good results for settings with high percentage of outliers. The informative variables were detected accurately. We illustrated the performance of SPRM on a data set of polymer stabilization formulations of moderate dimensionality and on high-dimensional gene expression data. An implementation of the SRPM, as well as visualization tools and the cross-validation model selection method outlined in Section 2.4, is available on CRAN in the package `sprm` (Serneels and Hoffmann, 2014).

The extension of SPRM regression for a multivariate response is a next step to take. Note that few papers combine sparseness and robustness for multivariate statistics, an exception is Croux et al. (2013) for principal component analysis. The development of prediction intervals around the SPRM prediction is another challenge left for future research. A bootstrap approach seems reasonable, but its validity remains to be investigated. Obtaining theoretical results on breakdown point or consistency of the model section is out of the scope of this paper. Few theoretical results are available in the PLS literature, and this only for the nonrobust and nonsparse case. In this paper we proposed and put into practice a new sparse and robust partial least squares method, which we believe to be valuable for data scientists confronted with prediction problems involving many predictors and noisy data.

Acknowledgments

This work is supported by BASF Corporation and the Austrain Science Fund (FWF), project P 26871-N20.

CHAPTER 3

Sparse and robust PLS for binary classification

Abstract: Partial robust M regression (PRM), as well as its sparse counterpart sparse partial robust M regression (SPRM), have been reported to be regression methods that foster a partial least squares alike interpretation, while having good robustness and efficiency properties, as well as a low computational cost. In this paper, the partial robust M discriminant analysis classifier (PRM-DA) is introduced, which consists of dimension reduction through an algorithm closely related to PRM and a consecutive robust discriminant analysis in the latent variable space. The method is further generalized to sparse PRM-DA (SPRM-DA) by introducing a sparsity penalty on the estimated direction vectors. Thereby, an intrinsic variable selection is achieved, which yields a better graphical interpretation of the results, as well as more precise coefficient estimates, in case the data contain uninformative variables. Both methods are robust against leverage points within each class, as well as against *adherence outliers* (points that have been assigned a wrong class label). A simulation study investigates the effect of outliers, wrong class labels and uninformative variables on the proposed methods and its classical PLS counterparts, and corroborates the robustness and sparsity claims. The utility of the methods is demonstrated on data from mass spectrometry analysis (TOF-SIMS) of meteorite samples.

Key words: Discriminant analysis, Partial least squares, Robustness, Supervised classification, Variable selection

3.1 Introduction

Partial Least Squares (PLS) (Wold et al., 2001), is a powerful and popular method for compressing high-dimensional data sets. Commonly it is applied to two data blocks (predictors and response) and projects the data onto a latent structure such that the squared covariance between the blocks is maximized. PLS can deal with a high number of variables p and small sample size n and it is not affected by multicollinearity. Furthermore, it is popular in applied sciences because of the relative ease with which results can be visualized and interpreted. The latent components and scores can be displayed in biplots, which support the interpretation of the model and the understanding of the multivariate data structure.

Many classification methods can only be applied to data with more observations than variables. PLS is a well established tool for effective dimension reduction in the classification setting. Nguyen and Rocke (2002) proposed a two step approach for binary classification based on PLS. First the class memberships are modelled as binary variables and are treated for the projection on the latent structure as if they were a continuous response. In the second step a standard classifier, e.g. Fisher's LDA, is applied to the transformed data in the low dimensional space. This method is here referred to as PLS-DA.

The feasibility of such approaches has been discussed by Kemsley (1996) and in more detail by Barker (2000) and Barker and Rayens (2003). They established the theoretical connection between PLS on binary response and classification and showed that PLS directions maximize the between group variance. PLS classification methods have been applied with considerable success in various scientific research areas, as well as in industry and production. They were used to analyze food quality with conventional sensory profiling data Rossini et al. (2012), to classify waste water pollution Sääksjärvi et al. (1989) and infrared spectra of olive oils and plant seeds Kemsley (1996). They have been used for tumor classification with micro-array data Pérez-Enciso and Tenenhaus (2003) and for fault diagnosis in chemical processes Chiang et al. (2000).

Nevertheless, these methods have their drawbacks Kettaneh et al. (2005). For experimental data two challenges arise frequently which will be addressed here, namely

outliers, i.e. samples which are not coherent with the general trend of the data, and uninformative variables, which contain no explanatory power for the response and which commonly appear in large quantity in high-dimensional data sets.

Contamination, defect of instruments or wrong assumptions about the distribution of the data, may lead to apparently unreasonable measurements in the samples. In classical PLS, outliers have a much higher influence on the model estimation than ordinary observations and thereby, they distort the model. To avoid this problem in the regression framework, various robust PLS methods have been developed (for an overview, see Filzmoser et al. (2009)). Partial Robust M regression (PRM) (Serneels et al., 2005) is among the most popular of these methods, for its trade-off between robustness and (statistical and computational) efficiency. It is robust with respect to leverage points (outliers in the predictor space) and vertical outliers (outliers in the response).

PRM-DA is presented here as a robust alternative to PLS-DA. It inherits the advantages of a PLS method, such as the ability to deal with high-dimensional data, multicollinearity and the possibility to illustrate the model in biplots for interpretation. Furthermore, PRM-DA is closely related to PRM regression and as such, has good robustness properties, a high statistical efficiency, and is computationally fast. Due to the data structure of classification problems, the PRM algorithm for regression cannot be directly applied to a binary response but needs specific modifications for the detection of outliers. This aspect is presented in Section 3.3.

Another problem of increasing importance is the extraction of relevant variables from the data set. Variables which do not provide information about the class membership add unnecessary uncertainty to the model. For data with a high percentage of uninformative variables, biplots become overloaded and a sound interpretation of the model becomes tedious or even impossible Kettaneh et al. (2005). These issues can be countered by sparse modeling. An overview of existing sparse methods in Chemometrics is given in Filzmoser et al. (2012). A sparse coefficient estimate is obtained by imposing a penalty term (e.g. the L_1 norm of the coefficient vector), thanks to which uninformative variables are excluded from the model. In case the data contain uninformative variables, sparsity improves model precision and a parsimonious model is easier to interpret. Chun and Keleş (2010) introduced a sparse PLS regression method, which was adapted by Chung and Keleş (2010) to the classification setting. Following this approach, the sparse and robust classifier SPRM-DA is introduced in Section 3.4, which performs intrinsic variable selection and is related to SPRM regression (Hoffmann et al., 2015).

For the selection of the optimal model, the number of PLS components and a sparsity

parameter are determined through K-fold cross validation. The procedure is described in Section 3.5. To demonstrate the performance of the methods, simulation studies are conducted in Section 3.6 and data examples from mass spectrometry are given in Section 3.7.

3.2 Projection onto latent structure for discriminant analysis

In the binary classification problem, the data consist of observations from two different populations, henceforth referred to as *group A* and *group B* . Let n_A and n_B denote the number of observations of groups A and B, respectively, and $n = n_A + n_B$. The data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ with p variables can be divided into two subsets $\boldsymbol{X}_A \in \mathbb{R}^{n_A \times p}$ and $\boldsymbol{X}_B \in \mathbb{R}^{n_B \times p}$ containing the observations of groups A and B. In order to disencumber notation, we assume without loss of generality that $\boldsymbol{X}_A$ form the first n_A rows of $\boldsymbol{X}$, followed by the observations $\boldsymbol{X}_B$ of group B.

The first step of PLS-DA, the projection onto latent structure, is methodically equivalent to that in PLS regression. The class memberships of the data are coded in the vector $\tilde{\boldsymbol{y}}$ with 1 for group A and -1 for group B. It is centred and scaled and further treated as if it were a continuous response, denoted by $\boldsymbol{y}$. Furthermore, assume that $\boldsymbol{X}$ is column-wise centred.

Dimension reduction is achieved by projection of the original variables onto a latent structure, such that the covariance between the projection of the predictors and the response is maximized. In detail, the *direction vector* $\boldsymbol{w}_h$ of a PLS model (also known as *weighting vector*) maximizes

$$\boldsymbol{w}_h = \underset{\boldsymbol{w}}{\operatorname{argmax}} \operatorname{cov}^2 (\boldsymbol{X}\boldsymbol{w}, \boldsymbol{y}) , \tag{3.1a}$$

for $h \in \{1, ..., H\}$ subject to

$$\| \boldsymbol{w}_h \| = 1 \quad \text{and} \quad \boldsymbol{w}_h^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w}_i = 0 \text{ for } 1 \leq i < h. \tag{3.1b}$$

The direction vectors form the columns of $\boldsymbol{W} \in \mathbb{R}^{p \times H}$ and define the *latent components* or scores $\boldsymbol{T} \in \mathbb{R}^{n \times H}$ as linear combinations of the data, i.e. $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$. Since $\boldsymbol{y}$ and $\boldsymbol{X}$ are centred, an estimate of the (squared) covariance in (3.1a) is

$$\operatorname{cov}^2 (\boldsymbol{X}\boldsymbol{w}, \boldsymbol{y}) = \left(\frac{1}{n-1} \boldsymbol{y}^T \boldsymbol{X} \boldsymbol{w} \right)^2 . \tag{3.2}$$

Many algorithms exist to solve the maximization problem (3.1a) with this standard estimator. One of the most prominent is the NIPALS algorithm (Wold, 1975), which will be used in what follows.

After the dimension reduction of the data to dimensionality $H < p$, a standard classifier can be applied to the scores $\boldsymbol{t}_i$, which are the rows of $\boldsymbol{T}$. Here a simple linear classification rule is used, Fisher's Linear Discriminant Analysis (LDA). It assumes equal covariance structure of both groups. The classical pooled within-groups covariance estimate is defined by

$$\begin{aligned} \hat{\boldsymbol{\Sigma}} &= \frac{1}{n-2} \sum_{k \in \{A,B\}} \sum_{i \in C_k} (\boldsymbol{t}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{t}_i - \hat{\boldsymbol{\mu}}_k)^T \qquad (3.3) \\ \text{with} \quad \hat{\boldsymbol{\mu}}_k &= \frac{1}{n_k} \sum_{i \in C_k} \boldsymbol{t}_i \quad \text{for} \quad k \in \{A, B\} \end{aligned}$$

where $C_A = \{1, ..., n_A\}$ is the index set for group A and $C_B = \{n_A + 1, ..., n\}$ for group B. Following the LDA decision rule, a new observation $\boldsymbol{x}$ is then assigned to that group $k \in \{A, B\}$ that has the largest value of the discriminant score

$$\delta_k(\boldsymbol{x}) = (\boldsymbol{x}^T \boldsymbol{W})^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k - \frac{1}{2} \hat{\boldsymbol{\mu}}_k^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k + \log(\pi_k), \qquad (3.4)$$

wherein π_k are the prior probabilities of group adherence, and $\pi_A + \pi_B = 1$.

Kemsley (1996) has shown that the first PLS direction maximizes the univariate between-groups variance. Hence, good group separation can be expected from PLS dimension reduction, which facilitates classification in the score space. A more detailed discussion on the relationship between LDA and PLS is given in Barker and Rayens (2003). It establishes the theoretical foundation to use PLS dimension reduction for classification methods.

3.3 Robust discriminant analysis with PRM

Outliers in the data distort model estimation and therefore, predictions. Hence, it is essential to verify whether outliers are present in the data and to control their influence. In regression analysis, two types of outliers are generally distinguished: leverage points (outliers in the $\boldsymbol{X}$ space) and vertical outliers (in the $\boldsymbol{y}$ space). Within the framework of discriminant analysis, we need to deal with leverage points separately for each group, since each population has its own data structure. For each group those samples are identified which have values beyond a certain threshold, given the (robustly estimated)

covariance structure of the data. The concept of vertical outliers in regression cannot be directly translated to discriminant analysis, since the response is a categorical variable. However, in practice, errors may occur in the encoding of the group membership. These cases are*adherence outliers*. We label observations as adherence ($\boldsymbol{y}$) outliers, if the supervised group membership, i.e. the class coded in $\boldsymbol{y}$, is intrinsically wrong. This can be assessed by evaluating its position in the estimated score space.

A powerful tool in robust statistics to identify and diminish the influence of outliers, is the concept of M-estimation. Weights between zero and one are assigned to each sample to regularize its influence on the model estimation, whereas weights smaller than one reduce the contribution of an observation to the estimation of the model parameters (and eventually, a zero weight excludes it). In Serneels et al. (2005) and Hoffmann et al. (2015), it has been described how the concept of M regression can be translated to the PLS regression problem. PLS regression, as well as PLS classification, consists of two steps, which both need to be robustified against the influence of outliers. The first step of PLS-DA is the dimension reduction by projection onto the latent structure. The direction of that projection may be distorted by outliers. In order to construct a robust method, case weights are used in the covariance maximization step (Eq. (3.2)). The data are then iteratively reweighted to find optimal weights. These weights are then also used to perform weighted, robust LDA in the score space. An overview of the algorithm is presented in Algorithm 3.1.

The initial weights are derived from the position of an observation within its group. In high-dimensions, the distances have less informative value since they get more and more similar with increasing dimensionality (Hall et al., 2005). Therefore, the weights are not directly obtained from the distances in the original space. Instead, group-wise PCA is used for dimension reduction, as it has been similarly applied in Filzmoser et al. (2008): The data are split into the two groups $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$, and each column is robustly scaled. Then a classical PCA model is estimated for each group, where the number of components H_k for $k \in \{A, B\}$ can be determined by, e.g., the broken stick rule [1], i.e. to retain the hth component if its eigenvalue is larger than $1/p \sum_{i=h}^{p} 1/i$. Since the data is scaled robustly, the classical variance is dominated by the outliers. Therefore, the first PCA components will highlight variables, which are important for the detection of outliers.

The squared Mahalanobis distance of an observation $\boldsymbol{x}$ with respect to a center $\boldsymbol{\mu}$

[1] Note that for these purposes, several alternative criteria could be applied. A good overview is given in (Jackson, 1993).

and covariance $\boldsymbol{\Sigma}$ is defined as

$$\mathrm{MD}^2(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}). \tag{3.5}$$

Outliers can be detected by the robust squared Mahalanobis distance in the PCA score space, here defined for group A,

$$\tilde{d}_i = \mathrm{MD}^2(\boldsymbol{t}_i^{PCA};\hat{\boldsymbol{\mu}}_A^{PCA},\hat{\boldsymbol{\Sigma}}_A^{PCA}) \quad \text{for } i=1,...,n_A. \tag{3.6}$$

It is the distance of the i-th PCA score vector $\boldsymbol{t}_i^{PCA}$ to $\hat{\boldsymbol{\mu}}_A^{PCA}$, the robust centre in the PCA score space of group A, given $\hat{\boldsymbol{\Sigma}}_A^{PCA}$, the robust covariance estimate of the PCA scores. Robust centre and covariance are determined by a fast, high breakdown joint estimate of location and covariance. Estimators suitable for these purposes are e.g. the MM estimator (M. Salibián-Barrera and Willems, 2006) or the Fast MCD algorithm (Rousseeuw and Driessen, 1999). The results shown in this article are computed using robust starting values based on Fast MCD. In the same way, distances for observations from group B are obtained. Then the distances used for outlier detection are

$$d_i = \frac{\tilde{d}_i}{\mathrm{med}_{j\in C_k}\,\tilde{d}_j}\chi^2_{H_k}(0.5) \quad \text{for } i\in C_k \text{ and } k\in\{A,B\}. \tag{3.7}$$

The robust squared Mahalanobis distance is approximately $\chi^2_{H_k}$ distributed with the dimension of the data as degrees of freedom if the majority of the data is normally distributed. By the transformation in (3.7), the median of d_i equals $\chi^2_{H_k}(0.5)$, the 0.5 quantile of the chi-squared distribution with H_k degrees of freedom equal to the number of principal components of the model of group k.

The initial weights are calculated from the distances d_i using Hampel's redescending weighting function

$$\omega_1(d) = \begin{cases} 1 & |d|\le Q_1 \\ \frac{Q_1}{|d|} & Q_1<|d|\le Q_2 \\ \frac{Q_3-d}{Q_3-Q_2}\frac{Q_1}{|d|} & \text{if} \quad Q_2<|d|\le Q_3 \\ 0 & Q_3<|d|. \end{cases} \tag{3.8}$$

A sensible choice for the parameters Q_1, Q_2 and Q_3 are the 0.95, 0.975 and 0.99 quantiles of the chi-squared distribution with as degrees of freedom the number of components used in the PCA model. Then the initial weights are $\omega_i = \omega_1(d_i)$ for $i=1,...,n$. A diagonal matrix $\boldsymbol{\Omega} = \mathrm{diag}(\omega_1,...,\omega_n)$ is used to downweight the observations. Let $\boldsymbol{X}_\Omega = \boldsymbol{\Omega}\boldsymbol{X}$ be the weighted data matrix, where every observation has been multiplied by its case

weight. The weighted response is denoted by $\boldsymbol{y}_\Omega = \boldsymbol{\Omega y}$. Then the PLS maximization criterion is solved for the weighted data,

$$\hat{\boldsymbol{w}}_h = \underset{\boldsymbol{w}}{\operatorname{argmax}} \operatorname{cov}^2(\boldsymbol{X}_\Omega \boldsymbol{w}, \boldsymbol{y}_\Omega), \tag{3.9a}$$

subject to

$$\| \hat{\boldsymbol{w}}_h \| = 1 \quad \text{and} \quad \hat{\boldsymbol{w}}_h^T \boldsymbol{X}_\Omega^T \boldsymbol{X}_\Omega \hat{\boldsymbol{w}}_i = 0 \text{ for } 1 \leq i < h. \tag{3.9b}$$

The actual maximum is found by applying the NIPALS algorithm to the weighted data.

Starting with the PLS model estimated from data weighted with the initial weights, the case weights are updated iteratively. The score matrix $\boldsymbol{T} = \boldsymbol{X}\hat{\boldsymbol{W}}$ is divided into $\boldsymbol{T}_A$ and $\boldsymbol{T}_B$ with scores which belong to group A and B, respectively. From these matrices, robust Mahalanobis distances are calculated with the fast MCD estimator and then transformed as in (3.7). As before, the weighting function is applied to these distances d_i and the weights obtained, are $w_i^t = \omega_1(d_i)$. In the algorithm for regression, the calculation of these weights is simplified, because the side constraint = 0 leads to uncorrelated scores. For the classification setting, it is important to consider the covariance structure of the groups.

To identify observations which have probably the wrong coding of the class membership, we use an LDA related approach. Barker and Rayens (2003) showed that for a classification problem with two groups, the first PLS direction is the direction that maximizes between group variance. Considering this property, we assume that projection on the first PLS component will lead to good group separation. Let $\boldsymbol{t}_1^{(s)}$ denote the vector of the group wise scaled (not centred) scores of the first component and let m denote the midpoint between the two robust group centers of $\boldsymbol{t}_1^{(s)}$. We use m as the point of separation. For each group the observations with values of $\boldsymbol{t}_1^{(s)}$ on the wrong side of m will be down-weighted. We define

$$\boldsymbol{v} = (\boldsymbol{t}_1^{(s)} - m\mathbf{1}_n)^T \tilde{\boldsymbol{y}}, \tag{3.10}$$

where $\tilde{\boldsymbol{y}}$ is the vector with the class memberships coded as 1 and -1, and $\mathbf{1}_n$ is a vector of ones. The entries v_i of $\boldsymbol{v}$ are negative for those observations for which the corresponding value of the vector $\boldsymbol{t}_1^{(s)}$ does not accord with the given class membership in $\tilde{\boldsymbol{y}}$. Values smaller than a negative threshold should be excluded from the model estimation, since the label in $\tilde{\boldsymbol{y}}$ may be incorrect. For this purpose we use a modified Tukey's Biweight

function

$$\omega_2(v) = \begin{cases} 0, & v \leq c \\ \left(1 - \left(\frac{v}{c}\right)^2\right)^2 & \text{if} \quad c < v \leq 0 \\ 1, & v > 0 \end{cases} \tag{3.11}$$

$$c := \begin{cases} N^{-1}(0.01) & \text{if } N^{-1}(0.01) < 0 \\ 0 & \text{else} \end{cases} \tag{3.12}$$

with the 0.01 quantile $N^{-1}(0.01)$ of the normal distribution $N(\text{med}(\boldsymbol{v}), 1)$. The weights are denoted by $w_i^y = \omega_2(v_i)$ for $i = 1, ..., n$.

The final case weights for the reweighting of the data are defined as

$$\omega_i = \sqrt{\omega_1(d_i)\,\omega_2(v_i)}. \tag{3.13}$$

For some situations the weights $\omega_2(v_i)$ are not reasonable, e.g. when the known class membership of the observations is reliable. Then only the robust distances within each group should be considered, i.e. $\omega_i = \omega_1(d_i)$. The data are weighted with the updated case weights $\boldsymbol{\Omega} = \text{diag}(\omega_1, ..., \omega_n)$. The reweighting of $\boldsymbol{X}$ by $\boldsymbol{X}_\Omega = \boldsymbol{\Omega X}$ is repeated till convergence of the case weights ω_i.

In the second step of PRM-DA, a robust linear classifier is applied to the scores $\boldsymbol{T} = \boldsymbol{X}\hat{\boldsymbol{W}}$. Robust estimates are plugged into the LDA decision rule described in (3.4) using the weights derived in the first step. They are defined by

$$\begin{aligned} \hat{\boldsymbol{\mu}}_k &= \frac{\sum_{i \in C_k} \omega_i \boldsymbol{t}_i}{\sum_{i \in C_k} \omega_i} \quad \text{for} \quad k \in \{A, B\} \quad \text{and} \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{(\sum_{i=1}^n \omega_i) - 2} \sum_{k \in \{A,B\}} \sum_{i \in C_k} \omega_i (\boldsymbol{t}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{t}_i - \hat{\boldsymbol{\mu}}_k)^T. \end{aligned} \tag{3.14}$$

as in Todorov and Pires (2007).

Algorithm 3.1: The PRM-DA algorithm

---

1. Calculate initial case weights:
   - Estimate for each group a PCA model with group-wise robustly scaled data.
   - Choose the number of components of the PCA models by the broken stick rule.

- Calculate the robust MD^2 (3.5) of the PCA scores for group A

$$\tilde{d}_i = \text{MD}^2(\boldsymbol{t}_i^{PCA}; \hat{\boldsymbol{\mu}}_A^{PCA}, \hat{\boldsymbol{\Sigma}}_A^{PCA}) \quad \text{for } i = 1, ..., n_A \tag{3.15}$$

  and analogous for group B.

- Distances are transformed to d_i as described in (3.7) and the initial case weights are defined by $\omega_i = \omega_1(d_i)$.

2. Centre data robustly about the column wise median: $\boldsymbol{X}$
   Centre and scale response with mean and standard deviation: $\boldsymbol{y}$

3. Reweighting process: Repeat until convergence of the case weights.

   - Weight data:

$$\begin{aligned} \boldsymbol{X}_\Omega &= \text{diag}(\omega_1, ..., \omega_n)\boldsymbol{X} \\ \boldsymbol{y}_\Omega &= \text{diag}(\omega_1, ..., \omega_n)\boldsymbol{y} \end{aligned}$$

   - Apply NIPALS algorithm for H components to $\boldsymbol{X}_\Omega$ and $\boldsymbol{y}_\Omega$ and obtain robust direction matrix $\boldsymbol{W}_\Omega$. Define scores $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}_\Omega$.

   - Calculate weights for outliers in the predictor space.
     - Split scores $\boldsymbol{T}$ into group A and B, denoted by $\boldsymbol{T}_A$ and $\boldsymbol{T}_B$.
     - Calculate the robust MD^2

$$\tilde{d}_i = \text{MD}^2(\boldsymbol{t}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \quad \text{for } i \in C_k \text{ and } k \in \{A, B\}. \tag{3.16}$$

     - $\omega_i^t = \omega_1(d_i)$ with

$$d_i = \frac{\tilde{d}_i}{\text{med}_{j \in C_k} \tilde{d}_j} \chi^2_H(0.5) \quad \text{for } i \in C_k \text{ and } k \in \{A, B\}, \tag{3.17}$$

       where $\chi^2_H(0.5)$ is the 0.5 quantile of the chi-square distribution with H degrees of freedom.

   - Calculate weights for potentially wrong class labels.
     - Robustly scale the first column of $\boldsymbol{T}$ group wise: $\boldsymbol{t}_1^{(s)}$
     - Let m be the midpoint between robust group centres of $\boldsymbol{t}_1^{(s)}$.
     - Define measure of group coherence

$$\boldsymbol{v} = (\boldsymbol{t}_1^{(s)} - m\boldsymbol{1}_n)\tilde{\boldsymbol{y}}$$

  - Define $\omega_i^y = \omega_2(v_i)$.
- Update case weights $\omega_i = \sqrt{\omega_i^t \omega_i^y}$.

4. Classify with LDA decision rule (3.4) in the score space based on robust estimates described in (3.14).

3.4 Sparse robust discriminant analysis with SPRM

Sparse models are constructed such that only certain variables contribute to the prediction. In PLS based models, sparsity can be achieved when complete rows of $\boldsymbol{W}$ are zero. Then the corresponding variables have no influence on the scores $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$.

Chun and Keleş (2010) introduced a sparse PLS regression method, which was extended by Chung and Keleş (2010) to the classification setting. The central idea is to penalize the estimation of the direction vector $\boldsymbol{w}_h$ by an L_1 norm penalty. To gain more flexibility in the estimation and therefore more sparsity in the model, a surrogate direction vector $\boldsymbol{c}$ is introduced and the PLS criterion (3.9a) for downweighted data, with the standard covariance estimator (3.2) as plug-in, is modified to:

$$\min_{\boldsymbol{c},\boldsymbol{w}} -\frac{1}{2}\left(\boldsymbol{y}_\Omega^T \boldsymbol{X}_\Omega \boldsymbol{w}\right)^2 + \frac{1}{2}\left(\boldsymbol{y}_\Omega^T \boldsymbol{X}_\Omega (\boldsymbol{c} - \boldsymbol{w})\right)^2 + \lambda_1 \|\boldsymbol{c}\|_1 \tag{3.18a}$$

subject to

$$\|\hat{\boldsymbol{w}}\| = 1 \quad \text{and} \quad \hat{\boldsymbol{w}}^T \boldsymbol{X}_\Omega^T \boldsymbol{X}_\Omega \boldsymbol{w}_i = 0 \quad \text{for } 1 \leq i < h. \tag{3.18b}$$

with $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{w}}$ are the vectors minimizing (3.18a). The final estimate of the direction vector is

$$\hat{\boldsymbol{w}}_h = \frac{\hat{\boldsymbol{c}}}{\|\hat{\boldsymbol{c}}\|} \quad \text{for } h = 1, ..., H. \tag{3.18c}$$

The parameter λ_1 is the sparsity parameter, which controls for the amount of zeros in $\hat{\boldsymbol{c}}$.

Algorithm 3.2: The sparse NIPALS algorithm

Let $\boldsymbol{X}$ denote a column wise centred matrix and $\boldsymbol{y}$ the centred response.
Define $\boldsymbol{E}_1 = \boldsymbol{X}$. For $h = 1, ..., H$:

- $\boldsymbol{z}_h = \boldsymbol{E}_h^T \boldsymbol{y} / \|\boldsymbol{E}_h^T \boldsymbol{y}\|$

- $\boldsymbol{v}_h = (|\boldsymbol{z}_h| - \eta \max_i |z_{ih}|) \odot \boldsymbol{I}(|\boldsymbol{z}_h| - \eta \max_i |z_{ih}| > 0) \odot \operatorname{sgn}(\boldsymbol{z}_h)$
- $\boldsymbol{t}_h = \boldsymbol{E}_h \boldsymbol{v}_h$
- $\boldsymbol{E}_{h+1} = \boldsymbol{E}_h - \boldsymbol{t}_h \boldsymbol{t}_h^T \boldsymbol{E}_h / \|\boldsymbol{t}_h\|^2$

where $\odot$ is the Hadamard (or element wise) matrix product. The weighting vectors $\boldsymbol{v}_h$ of the deflated matrix $\boldsymbol{E}_h$ form the columns of $\boldsymbol{V}$. Then the sparse PLS direction vectors for the transformation of $\boldsymbol{X}$ are defined by $\boldsymbol{W} = \boldsymbol{V}(\boldsymbol{V}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{V})^{-1}$ and the scores are $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$.

The minimization problem (3.18) has an exact solution Chun and Keleş (2010), thanks to which a sparse NIPALS algorithm can be constructed. In Algorithm 3.2 weighting vectors are penalized by a fraction $\eta \in [0, 1)$ of its largest entry. The expression $\eta \max_i |z_{ih}|$ replaces λ_1 to facilitate the parameter selection as described in Section 3.5 since the range of η is known. So the complexity of the models can be varied from the full model to a nearly empty model.

In Hoffmann et al. (2015) this approach was robustified for regression analysis. Here the related SPRM-DA algorithm for classification is introduced, which follows the steps described in Algorithm 3.1 for PRM-DA with the sparse NIPALS (see Algorithm 3.2) instead of the NIPALS.

3.5 Parameter selection

For PRM-DA models, the number of latent components H needs to be determined and for the sparse methods additionally the sparsity parameter η has to be specified (see Algorithm 3.2). K-fold cross validation is a common tool to decide for the model parameters. Thereunto, the samples are divided randomly into K subsets. Each subset is used once as test data, while the rest of the samples are the training data. For a fixed parameter combination H and η, the model is estimated on the training data and the class membership is predicted for the test data. To compare the predictions across different models, a robust cross validation criterion is introduced.

Since the predicted class membership of outliers is not reliable, its effect on the evaluation should be downweighted. Let $M_A := \{i : y_i = 1 \wedge \operatorname{sign}(\hat{y}_i) = -1\}$ denote the set of indices of misclassified observations from group A within the test data and $C_A := \{i : y_i = 1\}$ all indices from group A test data (analogous for group B). Within

each cross validation loop, weights are calculated for the test data according to their position in the estimated score space. Let $\omega_1, \ldots, \omega_n$ denote the resulting weights of all test observations. Then we define the robust misclassification rate as

$$rmcr = \frac{\left(\frac{\sum_{i \in M_A} \omega_i}{\sum_{i \in C_A} \omega_i} + \frac{\sum_{i \in M_B} \omega_i}{\sum_{i \in C_B} \omega_i} \right)}{2}. \tag{3.19}$$

The class membership of an observation with weight zero has no influence at all on this decision criterion, whereas the misclassification of an observation which is not considered as outlier has the largest influence. This reflects the idea that observations with increasing distance from the main data structure have diminishing influence on the choice of the model. The model with minimum robust misclassification rate is chosen as optimal model.

For data without outliers, i.e. weights equal to one, this criterion is the common misclassification rate, which gives equal importance to the correct classification of both groups, independent of their group size,

$$mcr = \frac{\left(\frac{c_A}{n_A} + \frac{c_B}{n_B} \right)}{2}, \tag{3.20}$$

where c_A and c_B denote the number of misclassified observations which belong to group A and B, respectively.

3.6 Simulation studies

We generate data coming from two groups under the assumption that the variables follow a latent structure. Therefore, let $\boldsymbol{D} \in \mathbb{R}^{n \times q}$ consist of a block $\boldsymbol{D}_A$ with $n_A = 60$ rows and a second block $\boldsymbol{D}_B$ with $n_B = 60$ rows coming from multivariate normal distributions with mean $(M, -M, ..., -M) \in \mathbb{R}^q$ and $(M, ..., M) \in \mathbb{R}^q$, respectively, and with equal covariance. The covariance matrix has a block structure with two uncorrelated blocks of equal size; the covariance between the variables of each block is 0.7 and each variable has variance 1. The size of M determines whether or how much the groups are overlapping. We set $M = 1$ and $q = 10$. Then $\boldsymbol{y}$, which consists of the group memberships, is defined as $y_i = 1$ for $i = 1, ..., n_A$ and $y_i = -1$ for $i = n_A + 1, ..., n$.

We set $H = 2$ and apply the NIPALS algorithm to $\boldsymbol{D}$ and $\boldsymbol{y}$ in order to obtain a direction matrix $\boldsymbol{A}$ and loadings $\boldsymbol{P}$. The scores are $\boldsymbol{T} = (\boldsymbol{D} - \hat{\boldsymbol{\mu}})\boldsymbol{A}$, where $\hat{\boldsymbol{\mu}}$ is the column wise estimated centre of $\boldsymbol{D}$. Then the generated data is given by

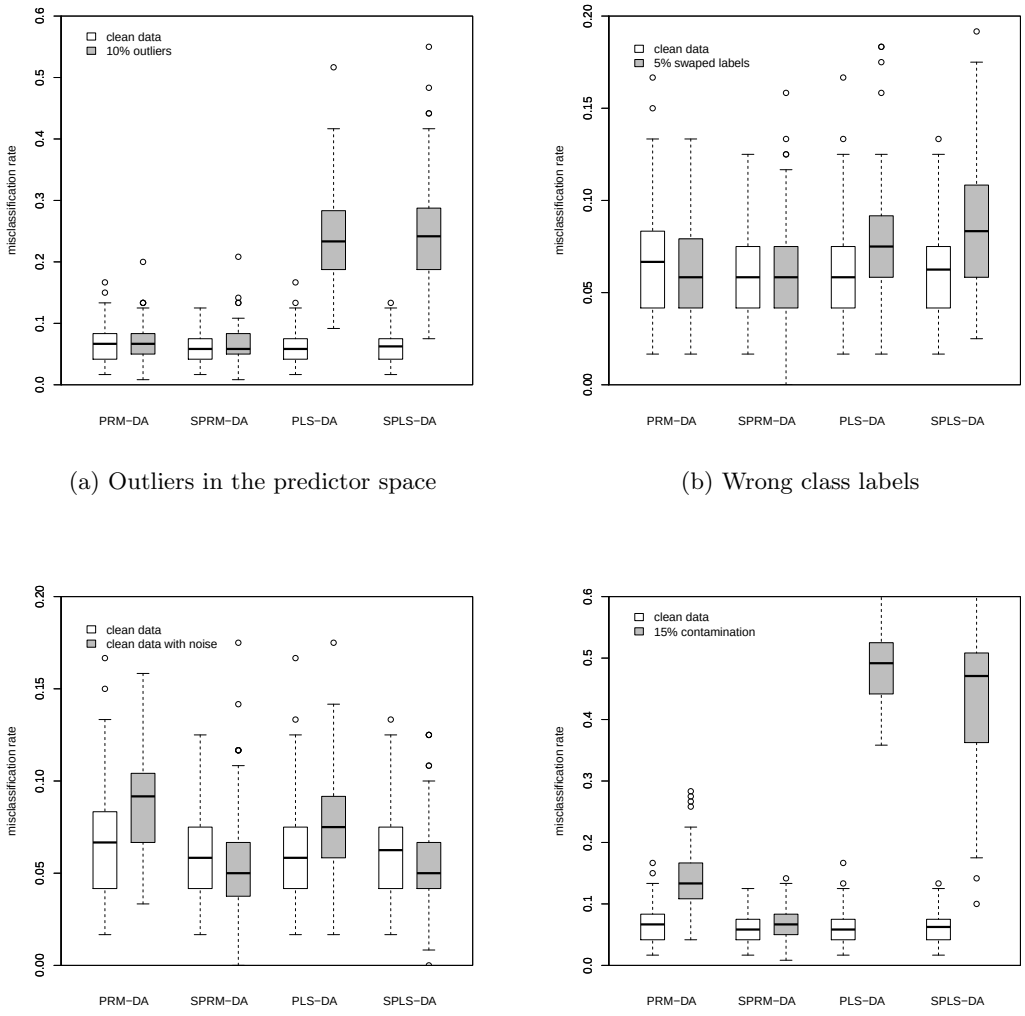
$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P} + \boldsymbol{E}, \tag{3.21}$$

where the values of $\boldsymbol{E} \in \mathbb{R}^{n \times p}$ come from the independent normal distribution $N(0, 0.2^2)$.

In this study, the data $\boldsymbol{X}$ is manipulated in three different ways to simulate common data problems: (i) Outliers in the predictor space: 10% outliers are generated by replacing the first $0.2n_A$ rows of $\boldsymbol{X}$ by independent values coming from a normal distribution with mean $(0, 10M, ..., 10M)$ and a diagonal covariance matrix with variance 0.1 for each variable. (ii) Wrong class labels: in group B, 10% of the group labels y_i are switched to 1. (iii) Uninformative variables: the rows of $\boldsymbol{TP} \in \mathbb{R}^{n \times q}$ are extended by values from a $p - q = 500$ dimensional normal distribution with zero mean, variances of one and covariances of 0.1. These 500 variables give no information about the class membership.

For the evaluation of the proposed methods, PRM-DA and SPRM-DA as well as their classical counterparts PLS-DA and SPLS-DA, training and test data are generated. The parameters are selected with 10-fold cross validation as described in Section 3.5, with choices of $H = 1, ..., 5$ and for the sparse methods $\eta = 0, 0.1, ..., 0.9$. For the selected parameters a model is estimated on the whole training data set. The model is then evaluated on the test data. Depending on the simulation setting the training data is contaminated with abnormal observations, i.e. outliers in the predictor space or wrong class labels. The test data is free from such contamination and so the accuracy of the classification model can be evaluated with the misclassification rate *mcr* defined in (3.20).

Figure 3.1 summarizes the results of the simulation study. The contamination in the predictor space leads to a heavy increase of the *mcr* for the classical methods (see Figure 3.1a). Also wrong class labels distort the classical methods, while no qualitative change in the *mcr* is visible for the robust methods (see Figure 3.1b). The limitations of PRM-DA and PLS-DA get visible when uninformative variables are added to the predictors (see Figure 3.1c). The effect of the combination of all three data problems is presented in Figure 3.1d. PLS-DA and SPLS-DA fail completely with a median misclassification rate of approximately 50%, which could have been obtained with equal likelihood from random group assignment. The median misclassification rate of PRM-DA does not represent reasonable models either and shows that the method is no longer robust to outliers in the presence of these 500 noise variables. The best results are obtained by SPRM-DA. While the interquartile range increases by the modification of the data, the median misclassification rate of SPRM-DA remains nearly the same.



(a) Outliers in the predictor space

(b) Wrong class labels

(c) Uninformative variables

(d) All modifications

Figure 3.1: Misclassification rate of test data.

3.7 Mass spectra of extraterrestrial material

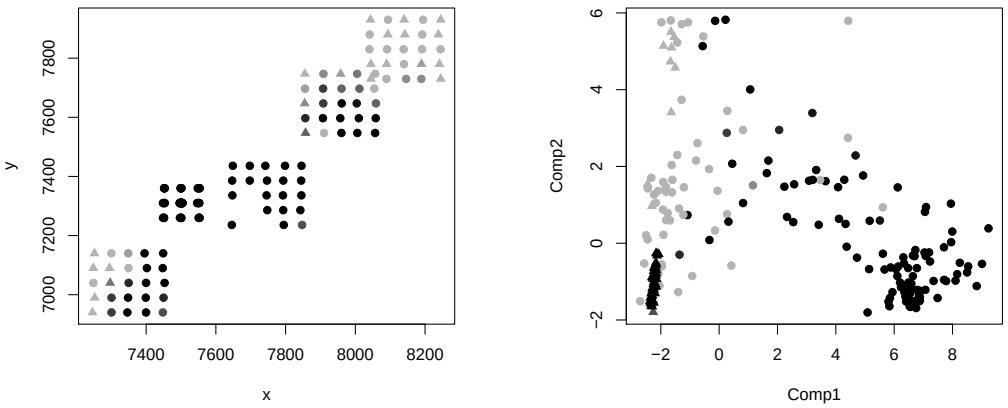
COSIMA (Kissel et al., 2007) is a TOF-SIMS (time-of-flight secondary ion mass spectrometry) instrument. It is on-board of ESA's Rosetta mission, where it collects dust particles of the comet Churyumov-Gerasimenko on gold or silver targets to study their chemical composition (Schulz et al., 2015). A twin laboratory instrument of COSIMA, located at Max Planck Institute for Solar System Research (Göttingen Germany), was used to analyze samples of meteorites from the Natural History Museum Vienna to support the analysis of the comet data.

One challenge is to identify the exact positions of comet dust particles on the target and to take measurements there. The spectra are typically obtained at rectangular grid positions located in the estimated area of the particles. The resulting data set consists of spectra taken on the grain as well as spectra from the background of the target.

We demonstrate the utility of the proposed methods for different research questions related to TOF-SIMS measurements on two meteorites (both prepared on the same target). One is meteorite Ochansk (observed fall 30 Aug 1887 near Perm, Russia), the other is meteorite Tieschitz (observed fall 15 Jul 1878 near Olomouc, Czech Republic); both are ordinary chondrites. The number of spectra used is 63 spectra from target background (gold), 155 spectra measured at or near an Ochansk particle, and 25 spectra measured at or near a Tieschitz particle. An original TOF-SIMS spectrum consists of the numbers of secondary ions in 30,222 flight-time bins for the mass range 0 to 400.52 mu. Preprocessing of the spectral data is briefly summarized as follows: mass range used 1 - 150 mu; only mass windows for inorganic ions are considered (Varmuza et al., 2011); signals from the primary ions ($^{115}In^{+}$) excluded; resulting in $p = 2612$ variables. Because qualitative aspects are of interest, the spectra were normalized to a constant sum (100) of the ion counts (rows in matrix $\boldsymbol{X}$).

Mislabelled data: TOF-SIMS spectra are measured across grids in the area where the material of interest is suspected. For the Ochansk measurements, visual inspection is possible to locate the grain, but the meteorite material may be spread invisibly in a larger area. At the edge of the meteorite, the spectra consist of a mixture of background and meteorite. For TOF-SIMS measurements of comet grains, it is difficult to locate the dust particles precisely and the recognition of potentially relevant spectra becomes especially important.

We split the spectra measured on and in the neighbourhood of the meteorite (group A) and background spectra (group B) randomly into five subsets, such that the sizes



(a) Predicted weights for potentially mislabelled samples (black: weight one) and class prediction (circle for grain group) given their position (x,y in μm) on the target.

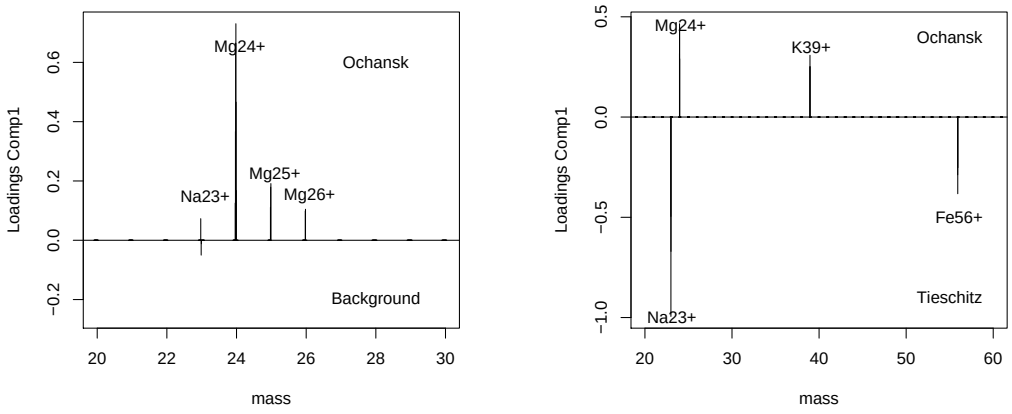
(b) Scores of the first two components of a SPRM-DA model for all Ochansk (circles) and background (triangles) data. The grey scores have case weights ω_i smaller than one.

Figure 3.2: SPRM-DA model for Ochansk and background spectra

of the groups across each split are approximately the same. Sequentially, each of the subsets is used as test data, while the rest is training data. An SPRM-DA model is estimated on the training data and the parameters of the model are chosen as described in Section 3.5 with 10 fold cross validation within the training set. Then the model is used to predict the class membership and to calculate weights of the test data.

To obtain a meaningful model, it is important for the estimation that spectra are used which were actually measured on the meteorite, i.e. that those spectra which come from the grain with a high probability have weight one. Class assignment of test samples to a group is only reliable for observations that are embedded in training data with weights equal to one. So one has to look jointly at weights and class prediction to gain more insight into the structure of the data and the meaning of the classification model.

The results for group A are shown in Figure 3.2a given the x- and y-coordinates of the measurements on the target. The weights for potentially mislabelled data ω_i^y are represented by the grey tone, black for weight one and continuously lighter grey for weights smaller than one. Small weights mean, that the corresponding sample is located close to the background samples. The area with black samples in Figure 3.2a coincides well with the area where the grain is visible on the target. All these spectra are



(a) Ochansk and background model

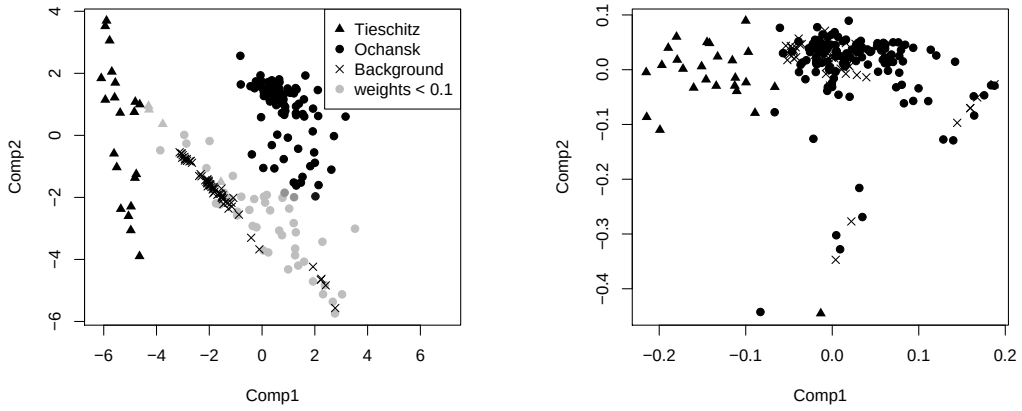
(b) Ochansk and Tieschitz model

Figure 3.3: Selected range of the loadings of the first component for SPRM-DA model.

predicted to belong to group A. It shows that this approach builds classification models based on the relevant data and by prediction of the weights also gives information about the applicability domain of the models.

For illustrative purposes, an SPRM-DA model is estimated for the complete data set. The parameters found with 10 fold cross validation are $H = 5$ and $\eta = 0.1$. The remaining number of variables (mass bins) in the model is 128. Figure 3.2b shows the scores of both groups for the first two components. From this two dimensional projection we can already see that samples from the meteorite group (circles) which are close to the background data (triangles) have small weights, i.e. are colored in light grey. Figure 3.3a shows that in the sparse loadings of the first component the magnesium isotopes are relevant for the separation between meteorite and background.

Outliers in the predictor space: Data from the meteorite Ochansk (group A) are compared to data of Tieschitz (group B). In this context the measurements on the two meteorite grains should form the two discriminant groups, while off grain measurements or other irregular data are considered as outliers. A pre-selection of grain spectra secures that the groups are not dominated by background spectra. Therefore, models of a meteorite grain and background data as described in the previous paragraphs are used to predict the class membership of the test data. Samples from the meteorite group that are assigned to the background group, are excluded. This leads to $n_A = 155$ and



(a) SPRM scores of Ochansk and Tieschitz - with background spectra projected into the score space.

(b) SPLS score plot of Ochansk and Tieschitz - with background spectra projected into the score space.

Figure 3.4: Score plots for models of Ochansk and Tieschitz.

$n_B = 25$. Due to the small group size we use 5 fold cross validation to choose the model parameters. They are $H = 2$ and $\eta = 0.2$, which leads to a model with 30 mass bins.

In spite of the pre-selection, we expect the main source of outliers to come from background measurements considered as meteorite spectra. To validate the model, the group of background spectra is projected into the SPRM score space. Figure 3.4a shows, that several Ochansk spectra (and one Tieschitz spectra) are located in the same area as the background spectra. Since the scores in this area receive weights smaller than 0.1, they are reasonably identified as outliers and they are not relevant for the model estimation. In comparison, Figure 3.4b shows the score plot for an SPLS model with two components. Background spectra projected into the score space are spread over the whole area of the Ochansk spectra, so that in the group of Ochansk no distinction between spectra measured on grain or off grain is possible.

The SPRM model separates the two ordinary chondrite meteorites well and the first component gives insight into the different elemental compositions of the two meteorites (see Figure 3.3b). Tieschitz has higher counts for sodium and iron and Ochansk for magnesium and potassium. This is also visible in the mean spectra of the two groups in Figure 3.5.

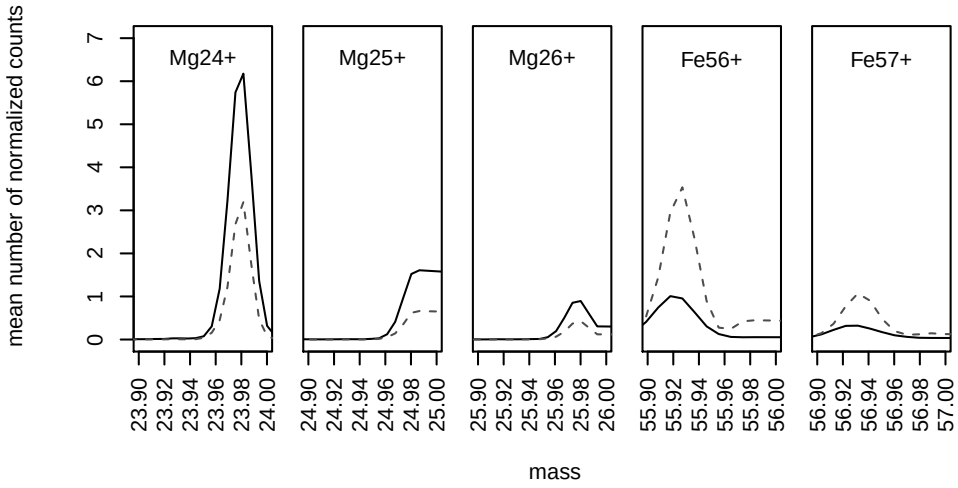


Figure 3.5: Selected mass ranges of mean spectra for Ochansk (black, solid line) and Tieschitz (grey, dashed line).

3.8 Conclusion

In this paper, a novel methodology for robust and when necessary, sparse, classification has been outlined. Several methods exist to estimate robust or sparse classification models, but to the best of our knowledge this is the first proposal of a sparse and robust method for binary classification. It inherits the visualisation and interpretation advantages that PLS-DA offers over many machine learning tools, the latter tendentiously yielding black box solutions. In contrast to classical PLS-DA, however, the new method is robust both to leverage points within each class, as well as to class adherence outliers. The method thanks its robustness essentially to a double pronged iterative reweighting scheme wrapped around the (sparse) NIPALS algorithm. Thereby, it is very germane to the earlier (sparse and non-sparse) partial robust M regression method for regression and has similar robustness and sparsity properties [2].

A simulation study has shown that outliers (leverage points and adherence outliers), as well as the presence of uninformative variables, can mislead PLS-DA and artificially inflate the misclassification rate. The new methods, on the contrary, still yield virtually unaffected misclassification performance in the presence of outliers. The sparse method (SPRM-DA) is the only method that also yields pristine performance when the data con-

[2]Implementations of these two methods have been made publicly available through the R package `sprm`, which can be downloaded through the CRAN network since 2014. Both new classification methods, as well as cross-validation and visualisation tools, have been appended to the same package in the latest version update (Serneels and Hoffmann, 2015).

tain both outliers and a non-negligible number of uninformative variables, even though also in this setting, PRM-DA still outperforms both classical methods, showing that the impact of outliers is the more harsh type of contamination studied. The simulations have also shown that for data without outliers, the performance of (S)PLS-DA and (S)PRM-DA is very similar. One would usually expect a slight advantage of the classical over the robust methods, in particular for very low sample sizes, because under normality the classical methods are known to be statistically more efficient than robust methods (Maronna et al., 2006). In practice, however, only the robust method allows to verify if outliers are present or not by investigating the case weights. The performance of SPRM-DA has been tested on a data set from meteorite samples, where it has largely managed to identify outliers and to classify samples according to the compositional classes they should belong to.

Acknowledgments

This work is supported by the Austrian Science Fund (FWF), project P 26871-N20. The authors thank F. Brandstätter, L. Ferrière, and C. Koeberl (Natural History Museum Vienna, Austria) for providing meteorite samples, C. Engrand (Centre de Sciences Nucléaires et de Sciences de la Matière, Orsay, France) for sample preparation, and M. Hilchenbach (Max Planck Institute for Solar System Research, Göttingen, Germany) for TOF-SIMS measurements.

CHAPTER 4

Robust and sparse multi-group classification by the optimal scoring approach

Abstract: We propose a robust and sparse classification method based on the optimal scoring approach. It is also applicable if the number of variables exceeds the number of observations. The data are first projected into a low-dimensional subspace according to an optimal scoring criterion. The projection only includes a subset of the original variables (sparse modeling) and is not distorted by outliers (robust modeling). In this low-dimensional subspace classification is performed by minimizing a robust Mahalanobis distance to the group centers. The low-dimensional representation of the data is also useful for visualization purposes. We discuss the algorithm for the proposed method in detail. A simulation study illustrates the properties of robust and sparse classification by optimal scoring compared to the non-robust and/or non-sparse alternative methods. Two real data applications are given.

Key words: High-dimensional data, Linear discriminant analysis, Penalization, Robustness, Supervised classification, Variable selection

4.1 Introduction

In linear discriminant analysis (LDA) the data originate from K different populations. The aim is to find linear decision boundaries to separate the observations from the K groups as well as possible and to predict the class membership of new, unlabeled observations. Several formulations for LDA exist. Fisher's approach to LDA searches for directions that maximize the between-group variance given the within-group variance. Equivalently, one can take the conditional class densities as multivariate normal with the same covariance matrix, and apply the Bayes classification rule. The formulation for LDA considered in this paper is optimal scoring (Hastie et al., 1994). It recasts the classification problem into a regression framework and models the class membership with a quantitative parameter for each class.

While these different approaches to LDA yield the same classification results (Johnson et al., 2002; Witten and Tibshirani, 2011) they are all limited to settings where $n > p$. Optimal scoring enables us to transfer new developments in high-dimensional regression analysis to the classification context. In regression analysis the problem of high-dimensional data, in particular data with more variables than observations, attracts a lot of attention. A variety of sparse methods have been developed. The best known is the Lasso regression estimate (Tibshirani, 2011). For a response $\boldsymbol{y}$ and a column-wise centered and scaled predictor matrix $\boldsymbol{X}$, it is defined as

$$\min_{\boldsymbol{\beta}} \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

for regression coefficients $\boldsymbol{\beta}$, where the L_1 norm is $\|\boldsymbol{a}\|_1 = \sum_{i=1}^{p} |a_i|$, for a vector $\boldsymbol{a} = (a_1, \ldots, a_p)^T$. Fast algorithms have been developed for Lasso regression (Efron et al., 2004; Wu and Lange, 2008). The Lasso shrinks several of the estimated regression coefficients to zero, and is therefore said to be *sparse*. The zero coefficients correspond to the variables that are not selected to be used in the model. Hence, the Lasso performs simultaneous model estimation and variable selection. The sparsity tuning parameter is λ, and increasing values of λ will favor more coefficients equal to zero which results in sparser models. This is especially useful for data sets including uninformative variables which do not contribute information to predicting the response. When uninformative variables are excluded, the precision of the estimation increases and the models are easier to interpret. Recently, Clemmensen et al. (2012) proposed a sparse version of multigroup LDA, by adding an L_1 penalty to the objective function of the optimal scoring problem. This leads to a sparse discriminant analysis method applicable for $n < p$ as well.

In this paper we propose a *robust* version of sparse optimal scoring. It is robust because it is resistant to outliers. It is well known that outliers may render a statistical method completely unreliable. This will not happen if a robust method is used. A variety of robust classification methods have been proposed (Hubert and Van Driessen, 2004; Todorov and Pires, 2007), but generally they are not applicable for data with $n < p$. Vanden Branden and Hubert (2005) proposed a robust classifier for high dimensions based on SIMCA, but it does not use sparse modeling, so all variables are included in the model. A sparse and robust classification method based on partial least squares was proposed by Hoffmann et al. (2016); however it was only for binary classification problems. Robust optimal scoring, even the non-sparse case, was not previously considered in the literature.

The paper is structured as follows. In Section 4.2, we review the optimal scoring approach to linear discriminant analysis. In Section 4.3, we introduce the proposed method and present the algorithm in detail. In Section 4.4, a strategy is outlined to select the sparsity tuning parameter. A simulation study competing with existing alternative methods is presented in Section 4.5. Illustrations using real-world examples are given in Section 4.6.

4.2 Optimal scoring for multigroup classification

We follow the notation of Clemmensen et al. (2012) to outline the optimal scoring method. Let $\boldsymbol{X}$ be the $n \times p$ data matrix with the observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ in its rows and $\boldsymbol{Y}$ an $n \times K$ matrix of dummy variables coding the class membership of the observations, i.e. $y_{ik} = 1$ if and only if observation $\boldsymbol{x}_i$ belongs to group k, and zero otherwise. The rows of $\boldsymbol{Y}$ are denoted by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$. The columns of $\boldsymbol{X}$ are centered to have mean zero and scaled to have unit variance. The aim of optimal scoring is to find projection vectors $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_H$, such that each $\boldsymbol{X}\hat{\boldsymbol{\beta}}_h$ is a good prediction of the corresponding vector $\boldsymbol{Y}\hat{\boldsymbol{\theta}}_h$, for $h = 1, \ldots, H$. The vector $\boldsymbol{Y}\hat{\boldsymbol{\theta}}_h$ then contains the scores of the group which each observation belongs to. The K components of the score vector $\hat{\boldsymbol{\theta}}_h$ are the numeric scores assigned to each of the groups. One takes H smaller than the number of groups K, commonly $H = K - 1$.

The projection vectors $\hat{\boldsymbol{\beta}}_h$ and the score vectors $\hat{\boldsymbol{\theta}}_h$ are obtained sequentially. Let $\boldsymbol{D} = \frac{1}{n}\boldsymbol{Y}^T\boldsymbol{Y}$ be a $K \times K$ diagonal matrix of class proportions. Set $\hat{\boldsymbol{\theta}}_0 = \mathbf{1}_K$, the K-vector

of ones. Then solve for $h = 1, \ldots, H$

$$\min_{\boldsymbol{\beta}_h, \boldsymbol{\theta}_h} \frac{1}{n} \|\boldsymbol{Y}\boldsymbol{\theta}_h - \boldsymbol{X}\boldsymbol{\beta}_h\|^2 \quad \text{s.t.} \quad \boldsymbol{\theta}_h^T \boldsymbol{D} \boldsymbol{\theta}_h = 1, \quad \boldsymbol{Q}_h^T \boldsymbol{D} \boldsymbol{\theta}_h = \boldsymbol{0},$$

where $\boldsymbol{Q}_h = [\boldsymbol{Q}_{h-1}, \hat{\boldsymbol{\theta}}_{h-1}]$ is a $K \times h$ matrix. Here, $\|.\|$ stands for the Euclidean norm.

The sparse optimal scoring method of Clemmensen et al. (2012) simply adds an L_1 penalty to the objective function.

$$\min_{\boldsymbol{\beta}_h, \boldsymbol{\theta}_h} \frac{1}{n} \|\boldsymbol{Y}\boldsymbol{\theta}_h - \boldsymbol{X}\boldsymbol{\beta}_h\|^2 + \lambda \|\boldsymbol{\beta}_h\|_1 \quad \text{s.t.} \quad \boldsymbol{\theta}_h^T \boldsymbol{D} \boldsymbol{\theta}_h = 1, \quad \boldsymbol{Q}_h^T \boldsymbol{D} \boldsymbol{\theta}_h = 0. \tag{4.1}$$

Estimators $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$ that solve (4.1) can be obtained iteratively. Starting with a random vector for $\boldsymbol{\theta}_h$, one computes the Lasso for $\boldsymbol{\beta}_h$. For a given $\boldsymbol{\beta}_h$ there exists an explicit solution of (4.1) for $\boldsymbol{\theta}_h$. One iterates further until convergence. For details, see Clemmensen et al. (2012).

Once the projection vectors are obtained, a standard LDA is performed in a low-dimensional space of dimension H. Let denote by $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ the projected observations in the rows of $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{B}$, $\boldsymbol{B} = [\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_H]$. Denote the group averages of the projected observations by $\boldsymbol{m}_k = \frac{1}{n_k} \sum_{i \in C_k} \boldsymbol{z}_i$, where C_k denotes the index set for observations from class k, and n_k is the number of observations in class k, for $k = 1, ..., K$. The within-group covariance matrix is

$$\boldsymbol{S} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i \in C_k} (\boldsymbol{z}_i - \boldsymbol{m}_k)(\boldsymbol{z}_i - \boldsymbol{m}_k)^T.$$

The Mahalanobis distance of an observation $\boldsymbol{z}$ to the center $\boldsymbol{m}_k$ is given by

$$MD(\boldsymbol{z}; \boldsymbol{m}_k, \boldsymbol{S}) = \left((\boldsymbol{z} - \boldsymbol{m}_k)^T \boldsymbol{S}^{-1} (\boldsymbol{z} - \boldsymbol{m}_k)\right)^{1/2}.$$

An observation $\boldsymbol{x}$, transformed to $\boldsymbol{z} = \boldsymbol{x}^T \boldsymbol{B}$, is then assigned to the class k with smallest value of

$$MD(\boldsymbol{z}; \boldsymbol{m}_k, \boldsymbol{S})^2 - 2 \log(\pi_k).$$

Here, π_k is the prior probability belonging to group k, with $\pi_1, \ldots, \pi_K = 1$. In the following, π_k is set to the class proportion of group k, so $\pi_k = n_k/n$.

4.3 Robust and sparse optimal scoring

We now propose an optimal scoring algorithm for data containing outliers and possibly more variables than observations. Furthermore, not all variables contribute information

about the class membership of the observations. We refer to these variables as uninformative variables. Our goal is to reduce the number of uninformative variables by sparse estimation. The data matrix $\boldsymbol{X}$ is robustly centered by the coordinate-wise median and each column is scaled by the median absolute deviation (MAD) (Hampel, 1974). The MAD is defined by $\text{MAD}(a_1, \ldots, a_n) = 1.48 \,\text{med}_i \, |a_i - \text{med}_j \, a_j|$ where 1.48 is a factor to get consistency at a normal distribution.

The aim is to reduce the influence of outlying observations on the model estimation. A common and powerful approach to achieve this in a regression model is the iteratively re-weighted least squares algorithm. Given a robust initial estimator, the influence of observations with large residuals is down-weighted by case weights. The coefficient estimates and the case weights are iteratively re-estimated. Here we will take a related approach.

Initial estimation

The vectors $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$ are estimated sequentially for $h = 1, \ldots, H$. As before, $\hat{\boldsymbol{\theta}}_0 = \mathbf{1}_K$. First, we obtain initial estimates for $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$. It is important that they are robust with regard to outliers and can be computed in high dimensions. These initial estimates are used to begin the iterative procedure to get the final $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$. Appendix 4.7 provides the full details for their computation.

Outlier weights

Residuals are computed as

$$r_i = \boldsymbol{y}_i^T \hat{\boldsymbol{\theta}}_h - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_h \quad \text{for } i = 1, ..., n.$$

The observations will be weighted so that potential outliers will receive less weight. The weights are calculated based on the residuals. Weights are calculated separately for each group. The robustly standardized residuals where we center by the median and scale by the MAD are denote $r_i^{(s)}$.

Hampel's re-descending weighting function (Hampel et al., 1986) is applied to the standardized residuals to obtain weights for each observation. This weighting function

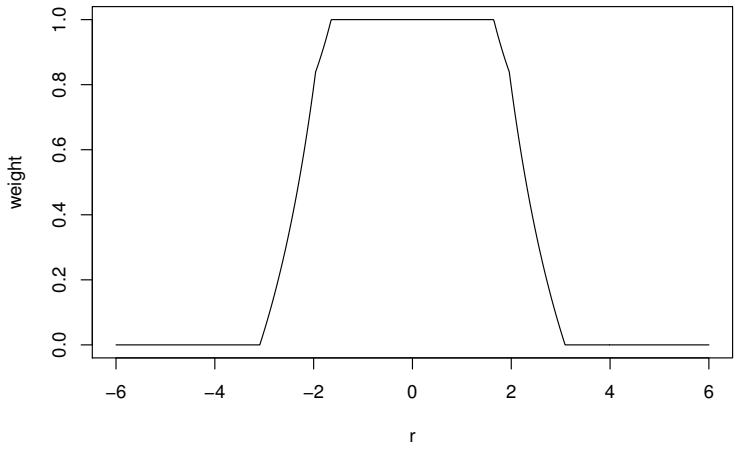


Figure 4.1: Hampel's re-descending weighting function.

is given as

$$\omega_H(r) = \begin{cases} 1 & |r| \leq q_1 \\ \frac{q_1}{|r|} & q_1 < |r| \leq q_2 \\ \frac{q_3 - r}{q_3 - q_2} \frac{q_1}{|r|} & \text{if} \quad q_2 < |r| \leq q_3 \\ 0 & q_3 < |r| \end{cases}$$

where the parameters q_1, q_2 and q_3 are set to the 0.95, 0.975 and 0.999 quantiles of the standard normal distribution, respectively (see Figure 4.1). The case weights are then $\omega_i = \omega_H(r_i^{(s)})$ for $i = 1, ..., n$. Under the assumption that the residuals are normal, 90% of the observations will receive the weight $\omega_i = 1$ and 0.2% will receive the weight $\omega_i = 0$.

Solving the weighted sparse optimal scoring problem

Let $\boldsymbol{\Omega}$ be a diagonal matrix with the case weights $\omega_1, ..., \omega_n$ on the diagonal. Then define the weighted data matrices as $\tilde{\boldsymbol{Y}} = \boldsymbol{\Omega}^{1/2}\boldsymbol{Y}$ and $\tilde{\boldsymbol{X}} = \boldsymbol{\Omega}^{1/2}\boldsymbol{X}$. The diagonal matrix $\tilde{\boldsymbol{D}} = \frac{1}{\sum \omega_i}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}$ contains on its diagonal the share of the total weight coming from each group's observations. The weighted sparse optimal scoring problem is defined as

$$\min_{\boldsymbol{\beta}_h, \boldsymbol{\theta}_h} \frac{1}{\sum \omega_i} \|\tilde{\boldsymbol{Y}}\boldsymbol{\theta}_h - \tilde{\boldsymbol{X}}\boldsymbol{\beta}_h\|^2 + \lambda\|\boldsymbol{\beta}_h\|_1 \quad \text{s.t.} \quad \boldsymbol{\theta}_h^T\tilde{\boldsymbol{D}}\boldsymbol{\theta}_h = 1, \quad \boldsymbol{Q}_h^T\tilde{\boldsymbol{D}}\boldsymbol{\theta}_l = \boldsymbol{0}. \tag{4.2}$$

If no outliers are detected, all weights are one, $\sum \omega_i = n$, $\boldsymbol{\Omega}$ is the identity matrix and Eq. (4.2) is the standard optimal scoring problem Eq. (4.1).

Equation (4.2) is solved by an alternating iterative scheme. For a given $\hat{\boldsymbol{\theta}}_h$, it reduces to the weighted Lasso regression problem

$$\hat{\boldsymbol{\beta}}_h = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{\sum \omega_i} \sum_{i=1}^{n} (\boldsymbol{y}_i^T \hat{\boldsymbol{\theta}}_h - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 \omega_i + \lambda \|\boldsymbol{\beta}\|_1. \tag{4.3}$$

which is equivalent to solving the Lasso for the weighted data, with the sparsity parameter given by $\lambda \sum w_i/n$. For a given $\hat{\boldsymbol{\beta}}_h$, the optimization problem Eq. (4.2) is solved by

$$\hat{\boldsymbol{\theta}}_h = c \left\{ \boldsymbol{I} - \boldsymbol{Q}_h (\boldsymbol{Q}_h^T \tilde{\boldsymbol{D}} \boldsymbol{Q}_h)^{-1} \boldsymbol{Q}_h^T \tilde{\boldsymbol{D}} \right\} (\tilde{\boldsymbol{D}}^{-1} \tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{X}} \hat{\boldsymbol{\beta}}_h) \tag{4.4}$$

where c is a scalar so that $\hat{\boldsymbol{\theta}}_h$ fulfills the side constraint $\hat{\boldsymbol{\theta}}_h^T \tilde{\boldsymbol{D}} \hat{\boldsymbol{\theta}}_h = 1$. The derivation of Eq. (4.4) is given in Appendix 4.7. Notice that the last part in parentheses in Eq. (4.4) is proportional to $(\tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{Y}})^{-1} \tilde{\boldsymbol{Y}}^T \tilde{\boldsymbol{X}} \hat{\boldsymbol{\beta}}_h$, the OLS estimate of $\boldsymbol{\theta}_h$ when regressing $\tilde{\boldsymbol{Y}}$ on $\tilde{\boldsymbol{X}} \hat{\boldsymbol{\beta}}_h$ without side constraints.

After computing $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$, new residuals r_i and case weights ω_i, for $i = 1, \ldots, n$, are calculated as described previously. New estimates of coefficient and score vectors are computed based on the re-weighted data as in Eq. (4.3) and Eq. (4.4).

Convergence criterion

Let $\omega_1^j, \ldots, \omega_n^j$ denote the case weights and $\hat{\boldsymbol{\beta}}_h^j$ and $\hat{\boldsymbol{\theta}}_h^j$ the estimates in the jth iteration step. Then the weighted mean residual sum of squares with Lasso penalty in the jth iteration step is

$$L_h^j = \sum_{i=1}^{n} (\boldsymbol{y}_i^T \hat{\boldsymbol{\theta}}_h^j - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_h^j)^2 \omega_i^j + \sum_{i=1}^{n} \omega_i^j \lambda \|\hat{\boldsymbol{\beta}}_h^j\|_1.$$

The convergence criterion for stopping the iterative procedure is defined as $|L_h^j - L_h^{j-1}|/L_h^j < 10^{-4}$.

Classification rule

The iterative procedure outlined in the previous subsections provides a projection matrix $\boldsymbol{B} = [\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_H]$. We project the data onto an H-dimensional subspace, i.e. $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{B}$, with the rows $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$. We observed that for a large sparsity parameter λ, the last column(s) of $\boldsymbol{B}$ may consist of only zeros. Then the dimension of the classification problem on the projected data is reduced automatically.

Instead of computing sample averages and covariance matrices of the projected data, we compute a robust location and covariance matrix estimator. For this, we take the

minimum covariance determinant (MCD) described in Rousseeuw and Driessen (1999). The robust group centers $\boldsymbol{m}_k$, for $k = 1, \ldots, K$, are the MCD location estimates of the projected observations from the kth group, i.e. of $\boldsymbol{z}_i$, $i \in C_k$. Then the projected observations are centered group-wise, $\tilde{\boldsymbol{z}}_i = \boldsymbol{z}_i - \boldsymbol{m}_k$ for $i \in C_k$ and $k = 1, ..., K$. A robust covariance estimate $\boldsymbol{S}$ from these pooled centered observations is obtained by the MCD covariance matrix estimate (Rousseeuw and Driessen, 1999). The decision rule for a new observation $\boldsymbol{x}$ is as follows: Project $\boldsymbol{x}$ onto the subspace, $\boldsymbol{z} = \boldsymbol{x}^T\boldsymbol{B}$ and compute the Mahalanobis distances to the group centers $\boldsymbol{m}_k$ with respect to $\boldsymbol{S}$. Assign $\boldsymbol{x}$ to group

$$\underset{k=1,...,K}{\operatorname{argmin}} (\boldsymbol{z} - \boldsymbol{m}_k)^T \boldsymbol{S}^{-1} (\boldsymbol{z} - \boldsymbol{m}_k) - 2\log(\pi_k).$$

4.4 Model selection and evaluation

Two steps are necessary for the proper evaluation of the proposed method. First, a strategy to select an optimal sparsity parameter is needed, second, the prediction performance for new observations is evaluated. We split the data randomly into calibration data and test data.

To select the optimal sparsity parameter λ^*, five-fold cross validation is performed on the calibration set. We split the calibration data randomly into $J = 5$ blocks of approximately equal size such that the observations from each class are evenly spread across the blocks. Each of the five blocks is used in turn as a validation set and the rest as a training set. For a sequence of values for the sparsity parameter $\lambda_1, \ldots, \lambda_L$ (covering the range between the full and the empty model) classification models are estimated using the training data and evaluated on the validation data. Since the decision for the optimal sparsity parameter λ^* should not be influenced by outliers, we propose using a weighted misclassification rate (wmcr) for evaluation. For the jth validation set, which consists of n_j observations $\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_{n_j}^j$ define

$$\text{wmcr}(\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_{n_j}^j, \lambda) = \frac{1}{K} \sum_{k=1,...,K} \frac{\sum_{i \in M_k^j(\lambda)} w_i^j(\lambda)}{\sum_{i \in C_k^j} w_i^j(\lambda)}, \tag{4.5}$$

where C_k^j is the index set of all observations from the validation set belonging to group k, and $M_k^j(\lambda)$ is the subset of C_k^j containing the indices of misclassified observations (for the model estimated using the sparsity parameter λ). The weight $w_i^j(\lambda)$ of an observation $\boldsymbol{x}_i^j$ is derived from the Mahalanobis distance to its closest center in the projected subspace,

i.e.

$$MD_i^j(\lambda) = \min_{k=1,\ldots,K} MD\left(\boldsymbol{x}_i^{jT}\boldsymbol{B}; \boldsymbol{m}_k, \boldsymbol{S}\right),$$

where $\boldsymbol{B}$, $\boldsymbol{m}_k$ and $\boldsymbol{S}$ are estimated on the jth training set with sparsity parameter λ. Then the weight is defined as

$$w_i^j(\lambda) = \begin{cases} 1 & MD_i^j(\lambda)^2 \leq \chi^2_H(0.975) \\ 1/MD_i^j(\lambda) & \text{else} \end{cases},$$

where $\chi^2_H(0.975)$ denotes the 97.5% quantile of the χ^2 distribution with H degrees of freedom. When all weights are equal to one, the wmcr is equivalent to the misclassification rate (mcr), the mean of the proportion of misclassified observations from each group.

The tuning parameter can now be selected such that the average wmcr for each of the $J = 5$ validation sets is minimized, i.e.

$$\tilde{\lambda} = \underset{\lambda \in \{\lambda_1,\ldots,\lambda_L\}}{\operatorname{argmin}} \frac{1}{J}\sum_j l^j(\lambda),$$

where, for easier notation, $l^j(\lambda) = \text{wmcr}(\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_{n_j}^j, \lambda)$, for $j = 1, \ldots, J$.

We then use the one-standard-error rule (Hastie et al., 2015): choose the model with the largest sparsity parameter such that its average wmcr is still within one standard error of the minimum average wmcr. Thus, the optimal sparsity parameter with the one-standard-error rule is

$$\lambda^* = \max\left\{\lambda \in \{\lambda, \ldots, \lambda_L\} \mid \frac{1}{J}\sum_{j=1}^{J} l^j(\lambda) < \frac{1}{J}\sum_{j=1}^{J} l^j(\tilde{\lambda}) + \text{se}(l^1(\tilde{\lambda}), \ldots, l^J(\tilde{\lambda}))\right\},$$

where $\text{se}(a_1, \ldots, a_J) = \sqrt{\text{var}(a_1, \ldots, a_J)/J}$ denotes the standard error. This strategy favors more parsimonious models and is a safeguard against over-fitting. With the optimal sparsity parameter λ^* the final model is estimated using the whole of the calibration data, and we obtain $\boldsymbol{B}$, $\boldsymbol{m}_k$ and $\boldsymbol{S}$.

For the evaluation of the model, the class memberships of test data are predicted. Since the evaluation should not be distorted by outliers in the test data, we use the weighted misclassification rate from Eq. (4.5). In the simulation study, test data are generated without outliers and all weights are set to equal one.

4.5 Simulation study

Simulation schemes: The data are generated from $K = 3$ different p-dimensional normal distributions representing three groups. The distributions have equal covariance structure, but different mean vectors. For group k ($k = 1, 2, 3$), let the mean be a vector of length p with value 2 for the kth variable and zeros elsewhere. So, the number of informative variables is $q = 3$. The diagonal of the covariance matrix is a vector of ones. The covariance between the informative variables is 0.1 and zero between all others. The number of observations is $n = 120$, where each group consists of $n_k = 40$ observations.

In the first scenario of this simulation study, the effect of increasing $p \in \{3, 13, 23, 53, 103, 203\}$ is illustrated, i.e. of increasing the number of uninformative variables while the number of informative variables $q = 3$ is fixed. The second scenario shows the effect of outliers on the methods, also for increasing $p \in \{3, 13, 23, 53, 103, 203\}$. Outliers are included in the calibration data by taking 10% of the observations of the first group and replacing their values for the first variable by random values from $N(-10, 1)$. Hence, there are still two uncontaminated informative variables. Finally, in a third scenario, the number of uninformative variables is set to 50 and the third informative variable is removed, i.e. $p = 52$. Outliers are again only generated in the first group by replacing the values of the first variable by random values from $N(-20, 1)$. This setting is more challenging because only one uncontaminated informative variable remains, and because the outliers take on more extreme values. The percentage of outliers in the first group ranges from 0% to 45% in increments of 5%, allowing us to observe the influence of increasing levels of contamination.

Methods and evaluation: The results from robust sparse optimal scoring (rSOS) and classical sparse optimal scoring (cSOS) are compared. For settings where non-sparse classification methods can be applied (i.e. $n > p$), models are estimated with LDA and robust LDA (rLDA). The latter method uses the MCD of the pooled centered data as the robust covariance matrix estimator, where the centers of each group are estimated by the location MCD estimator, as outlined in Hubert and Van Driessen (2004). Recall that LDA is equivalent to classical optimal scoring. to create a benchmark, we first remove all uninformative variables and outliers from the calibration data and then apply LDA. This benchmark method cannot be applied in practice, since one does not know which variables are informative and which observations are outliers. We refer to this method as the oracle; it gives an estimate of the lower boundary for the best misclassification rate we can achieve with linear boundaries.

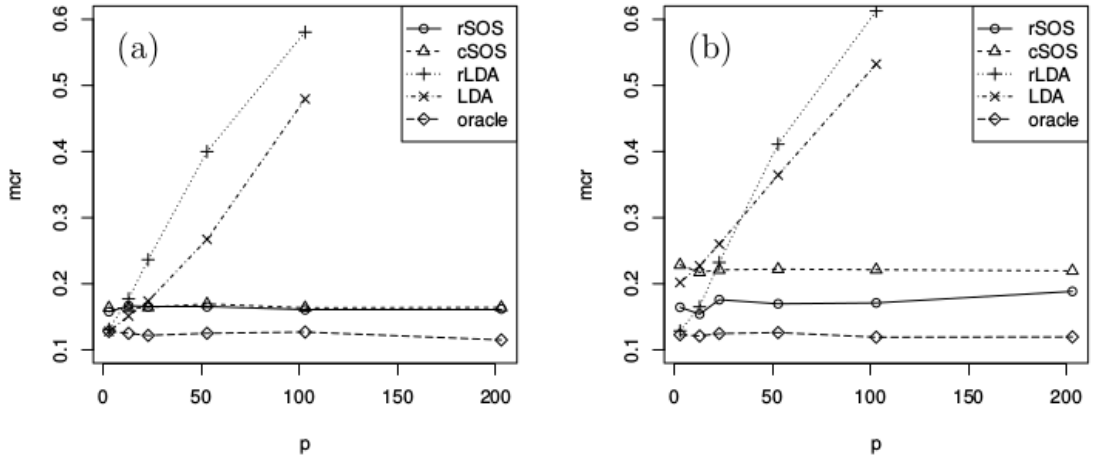


Figure 4.2: Misclassification rate (mcr) averaged over 100 simulation runs as a function of p, the number of variables. (a) Scenario 1: models estimated on clean calibration data; (b) Scenario 2: models estimated on calibration data with 10% outliers in one group.

For robust and classical sparse optimal scoring the sparsity parameter λ is selected with five-fold cross validation using the calibration data ($n = 120$) from a grid of values between 0.1 and 2 with a increment size 0.05, as described in Section 4.4.

To evaluate the models, test data of size $n = 120$ are generated in the same way as the calibration data, but without outliers for all scenarios. The predicted class membership of the test data is compared to the known, true class membership and the misclassification rate (mcr) is reported. Other quality criteria of the model concern the number of correctly selected variables. The false negative rate (FNR) is the fraction of informative variables not included in the model, the false positive rate (FPR) refers to the fraction of uninformative variables included in the model.

Simulation results: The results from the first scenario demonstrate the advantage of sparse modeling when the number of uninformative variables increases. Figure 4.2 (a) shows the misclassification rate for test data, averaged over 100 simulation runs. The benchmark mcr for this simulation design is about 12.5%, as can be seen from the results of the oracle. Hardly any difference between the performance of cSOS and rSOS is visible in this setting. The mcr of both methods remains stable despite an increasing number of uninformative variables. In very low dimensions, for instance $p = 3$, LDA and rLDA slightly outperform cSOS and rSOS, but when p increases, LDA and rLDA quickly break down and give bad classification results, even when p is still smaller than

n. This shows that excluding uninformative variables is crucial for the quality of the prediction performance.

Table 4.1 (a) shows the quality of the variable selection for cSOS and rSOS. The false negative rate is slightly higher for cSOS whereas the false positive rate is slightly higher for rSOS. Overall, both rates are low for both methods, which implies that the variable selection with the L_1 penalty achieves good results.

In the second scenario, the effect of 10% outliers is investigated, as illustrated in Figure 4.2 (b). The benchmark given by the oracle is again about 12.5%, as expected. For $p = 3$, the robust methods rLDA and rSOS outperform the classical methods LDA and cSOS. Increasing the number of variables heavily affects both LDA rLDA. The method which performs best is rSOS, since it can cope with both increasing dimensions and outliers. Note that for cSOS, the presence of outliers substantially increases the mcr, but the number of uninformative variables has no further notable effect; for rSOS, the mean mcr slightly increases when p approaches its highest value.

Table 4.1 (b) shows that cSOS fails to identify the informative variables in presence of outliers. The FNR of cSOS is around 33% in this scenario, since the first of the three informative variables, the contaminated one, is not included in the model anymore. In this scenario, the variables selected by cSOS do not contain any outliers, but since the information present in the first variable is lost, it still leads to an increased mcr. With

(a) Scenario 1

p	3	13	23	53	103	203
FNR cSOS	0.02	0.02	0.02	0.04	0.01	0.04
FNR rSOS	0.01	0.02	0.02	0.02	0.00	0.02
FPR cSOS		0.02	0.02	0.01	0.01	0.00
FPR rSOS		0.04	0.03	0.02	0.01	0.02

(b) Scenario 2

p	3	13	23	53	103	203
FNR cSOS	0.32	0.32	0.32	0.33	0.33	0.33
FNR rSOS	0.05	0.03	0.07	0.05	0.06	0.08
FPR cSOS		0.02	0.02	0.00	0.00	0.00
FPR rSOS		0.07	0.05	0.03	0.04	0.02

Table 4.1: Variable selection: the false negative rate (FNR) and the false positive rate (FPR) is averaged over 100 simulation runs for classical and for robust SOS for (a) Scenario 1: models estimated on clean calibration data; (b) Scenario 2: models estimated on calibration data with 10% outliers in one group.

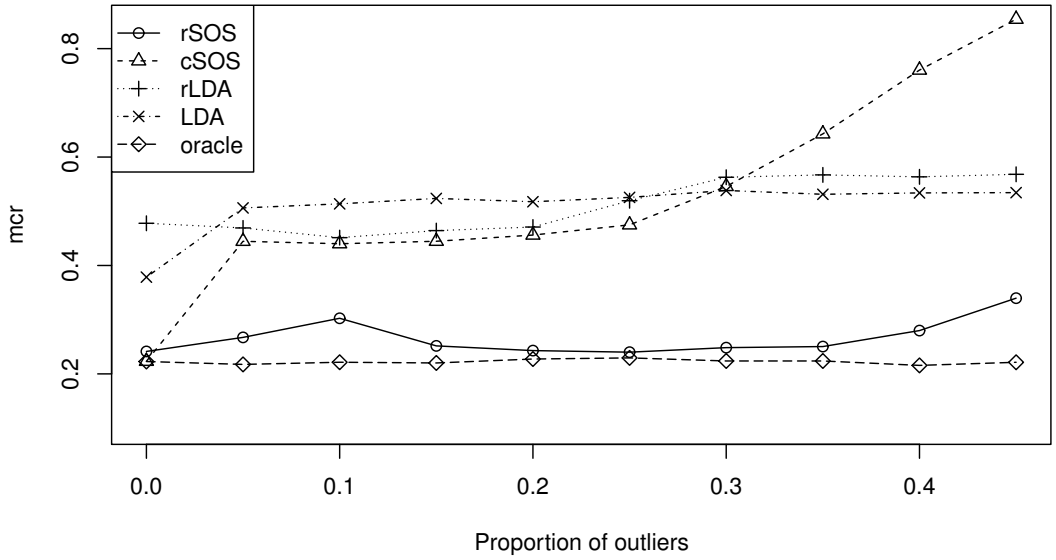


Figure 4.3: Scenario 3: Misclassification rate (mcr) averaged over 100 simulation runs as a function of increasing outlier proportion; $p = 52$.

rSOS, this information can be recovered; rSOS down-weights the outliers and is able to reveal that this first variable contributes enough information to be selected. Comparing the FNR of rSOS in Table 4.1 (a) and (b) shows an increase in the setting with outliers, but considerably smaller compared with cSOS. Finally, note that the FPR for rSOS is low, but slightly higher than for cSOS. In the second scenario, rSOS selects 4.7 variables on average, which is somewhat more than the average of 2.2 variables for cSOS.

Scenario three illustrates how the percentage of outliers influences the classification performance of the different methods. Figure 4.3 pictures the mcr as a function of the proportion of outliers in the calibration data, for $p = 52$. The benchmark given by the oracle is about 22.2% and indicates a lower boundary for the mcr. When there are no outliers, cSOS and rSOS are close to the oracle. However, as soon as there are only 5% outliers, the cSOS is strongly affected in its prediction performance, whereas rSOS continues to give reasonable results for larger percentages of outliers. As expected, the mcr of LDA and rLDA is inflated due to the $p - q = 50$ uninformative variables resulting in a high mean mcr, which increases slightly for higher percentages of contamination.

Computations are performed in R (R Core Team, 2016). For classical sparse optimal scoring, R code is available in the package `sparseLDA` (Clemmensen and Kuhn, 2012).

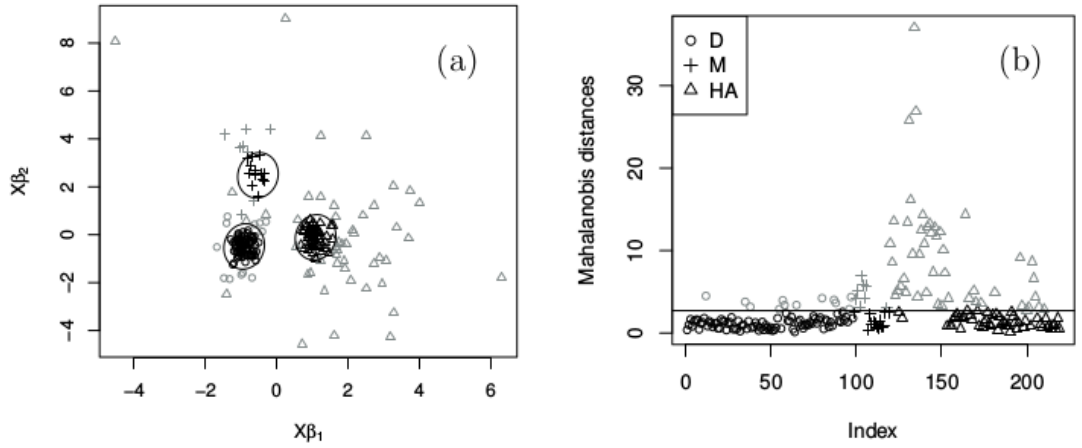


Figure 4.4: Fruit data: (a) visualization of 219 test observations in the projected subspace (b) Mahalanobis distance of each projected test observation to its group center. Observations with weights smaller than one are colored in gray.

The code for robust sparse optimal scoring is included in the package `rrcovHD` (Todorov, 2016).

4.6 Examples

Fruit data: This data set consists of $n = 1095$ measurements with $p = 256$ wavelengths from $K = 3$ different cultivars of the cantaloupe melon, named D, M and HA. We have 490 measurements from group D, 106 from group M and 499 from group HA. It is a well known benchmark data set to demonstrate the stability of robust classification methods (Hubert et al., 2008; Hubert and Van Driessen, 2004; Vanden Branden and Hubert, 2005). From former analyses it is known that the change of illumination led to outliers.

The data are split into calibration and test data (80% versus 20%) 5 times, such that all observations are included once in the test data and the observations from each class are evenly distributed across the test sets. For each calibration data set, the optimal sparsity parameter λ^* is selected as described in Section 4.4. We select it from a fine

	cSOS	rSOS
average mcr	0.028 (.0062)	0.041 (.0068)
average wmcr	0.016 (.0116)	0.009 (.0029)

Table 4.2: Fruit data: average (w)mcr is the (weighted) mcr averaged over the five test data sets. Standard errors are reported in parentheses.

grid starting with 10^{-4} up to 10^{-1} with increment size 0.002, covering model sizes from nearly full to empty. The weighted misclassification rate wmcr is calculated from the test data as in Eq. (4.5). The weights from the rSOS model are also used to calculate the wmcr for cSOS. Thereby observations which are detected as outliers by the rSOS model have small influence on the wmcr of cSOS.

The procedure is repeated for all calibration and test sets. The results are summarized in Table 4.2. It shows that the mcr of the cSOS is smaller than the mcr of the rSOS. On the other hand, the wmcr has a lower value for rSOS than for cSOS. The classical method tries to model the outliers, and since outliers are present in the test data, it achieves better results as well. The robust method, on the other hand, mainly models the non-outliers. So the weighted misclassification rate, which excludes the outliers, is lower for rSOS than for cSOS.

To visually depict the results, we randomly select one of the five data splits and apply rSOS to the calibration data. Figure 4.4 (a) shows the test observations projected onto the subspace. The ellipses defined by the sets $\{\boldsymbol{z} \in \mathbb{R}^2 | \mathrm{MD}(\boldsymbol{z}; \boldsymbol{m}_k, \boldsymbol{S}) = \sqrt{\chi^2_2(0.975)}\}$, for $k = 1, 2, 3$ enclose those observations which are considered non-outliers and which did receive weight one in the wmcr. The observations outside of the ellipses are colored in gray. Most outliers are from group HA which is in line with previous analyses (Vanden Branden and Hubert, 2005). In Figure 4.4 (b) the Mahalanobis distances of each test observation to its group center are shown. The horizontal line represents the cut-off value $\sqrt{\chi^2_2(0.975)}$. Again we see that many observations of HA have a large Mahalanobis distance in the projected space. Figure 4.4 (b) pinpoints other anomalous observations in all three groups.

Olive oil data: The data set *olitos* (Armanino et al., 1989) available in the R package `rrcovHD` (Todorov, 2016) contains $n = 120$ measurements on olive oil samples with $p = 25$ variables from fatty acids, sterols and triterpenic alcohols. The olive oils originate from Tuscany in Italy and are grouped into $K = 4$ classes representing different regions of production with group size 50, 25, 34 and 11. In this example, the number of variables is quite low, but rSOS can still be an appropriate method. We will compare its results to cSOS as well as to LDA and rLDA. To estimate and evaluate the models, the same approach is taken as described previously for the fruit data. The optimal sparsity parameter is searched on a grid from 0.01 to 1 with step size 0.05, which covers various model sizes from the full model to the empty model.

Table 4.3 summarizes the quality of the resulting models. With an average wmcr

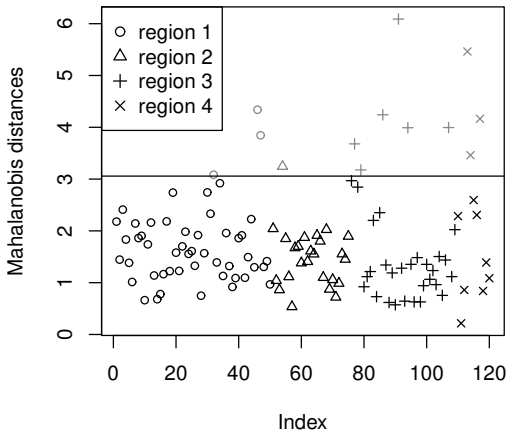


Figure 4.5: Olive oil data: Mahalanobis distances from rSOS of each projected observation to its group center. Observations with weights smaller than one are colored in gray.

of 13.3%, our proposed method rSOS performs better on this data set than the other methods. An interesting additional finding is that the mcr is lowest for rSOS with 15.3%. This may happen if there is no pattern in the outlier configuration.

The classical sparse method cSOS outperforms LDA slightly, and robust LDA has a much lower prediction quality than all other methods. Figure 4.5 shows the Mahalanobis distances from the rSOS estimator of the projected test data. Especially regions 3 and 4 have some observations with large distances to its group centers in the projected space. In Figure 4.6, the projection of all observations into the subspace is visualized.

4.7 Conclusion

This paper introduces a robust and sparse optimal scoring method for multigroup classification. It yields a new supervised classification method, applicable if the number of variables is large with respect to the sample size *and* with the possible presence of

	LDA	rLDA	cSOS	rSOS
average mcr	0.178 (.0645)	0.358 (.1460)	0.175 (.0308)	0.153 (.0312)
average wmcr	0.183 (.0604)	0.353 (.1201)	0.175 (.0431)	0.133 (.0282)

Table 4.3: Olive oil data: average (w)mcr is the (weighted) mcr averaged over the five test data sets. Standard errors are reported in parentheses.

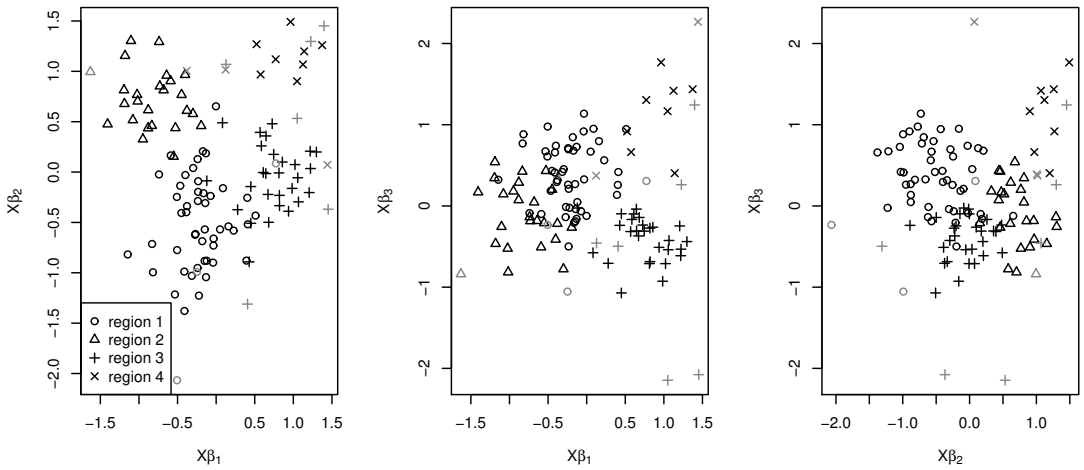


Figure 4.6: Olive oil data: pairwise scatter plots of data projected into the 3-dimensional subspace derived from rSOS. Observations with weight smaller than one are colored in gray.

outliers in the data. Using an iterative algorithm, it searches for an optimal projection into a subspace using only a subset of the original variables: the most informative ones. Potential outliers are down-weighted, reducing their influence in the search for this optimal projection. The final classification is then carried out in this $(K-1)$-dimensional subspace. As shown in the examples in Section 4.6, the resulting low-dimensional representation of the data is also useful for visualization and interpretation.

The algorithm we developed, which is outlined in Section 4.3, is implemented and publicly available in the R-package `rrcovHD` (Todorov, 2016). This package contains outlier detection methods and robust statistical procedures for high dimensions. A call to the function `SosDiscrRobust`, with the data matrix and the class memberships as input, returns the estimated model.

Only few proposals exist so far for robust classification in high dimensions. Our proposal has the important feature of being sparse, simultaneously performing variable selection and model estimation, by using a (robust) Lasso-type approach. The simulation study has shown the importance of considering both sparse modeling *and* robust estimation. If either of them is missing, the prediction performance may decrease drastically.

Acknowledgments

This work is supported by the Austrian Science Fund (FWF), project P 26871-N20.

Appendix

Derivation of expression (4.4) for the score vector estimates

Let $\omega_1, ..., \omega_n$ be case weights for each observation. $\boldsymbol{\Omega}$ is a diagonal matrix with these case weights in the diagonal. Then the weighted data matrices are $\tilde{\boldsymbol{Y}} = \boldsymbol{\Omega}^{1/2}\boldsymbol{Y}$ and $\tilde{\boldsymbol{X}} = \boldsymbol{\Omega}^{1/2}\boldsymbol{X}$. The diagonal matrix with weighted class proportions is $\tilde{\boldsymbol{D}} = \frac{1}{\sum \omega_i}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}$. The optimization problem (4.2) in step h for a given $\hat{\boldsymbol{\beta}}$ can be rewritten as

$$\min_{\boldsymbol{\theta}} \|\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{Y}}\boldsymbol{\theta}\|^2 \quad \text{s.t.} \quad \boldsymbol{\theta}^T\tilde{\boldsymbol{D}}\boldsymbol{\theta} = 1 \text{ and } \boldsymbol{C}\boldsymbol{\theta} = \boldsymbol{0} \in \mathbb{R}^h \tag{4.6}$$

with $\boldsymbol{C} = \boldsymbol{Q}^T\tilde{\boldsymbol{D}}$, and we drop the index h for ease of notation.

We use the method of Lagrange multipliers. The Lagrangian associated with Eq. (4.6) is given by

$$L = (\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{Y}}\boldsymbol{\theta})^T(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{Y}}\boldsymbol{\theta}) - \eta(\boldsymbol{\theta}^T\tilde{\boldsymbol{D}}\boldsymbol{\theta} - 1) - 2\boldsymbol{\gamma}^T\boldsymbol{C}\boldsymbol{\theta}.$$

The partial derivative set to zero gives

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -2\tilde{\boldsymbol{Y}}^T(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{Y}}\boldsymbol{\theta}) - 2\eta\tilde{\boldsymbol{D}}\boldsymbol{\theta} - 2\boldsymbol{C}^T\boldsymbol{\gamma} = \boldsymbol{0}.$$

Hence,

$$\boldsymbol{\theta} = (\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} + \boldsymbol{C}^T\boldsymbol{\gamma}).$$

To solve for the Lagrange multipliers η and $\boldsymbol{\gamma}$, the side constraints are used.

$$0 = \boldsymbol{C}\boldsymbol{\theta} = \boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} + \boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\boldsymbol{C}^T\boldsymbol{\gamma}$$

So,

$$\boldsymbol{\gamma} = -\left(\boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\boldsymbol{C}^T\right)^{-1}\boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}.$$

We conclude

$$\boldsymbol{\theta} = (\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1} \left\{\boldsymbol{I} - \boldsymbol{C}^T(\boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\boldsymbol{C}^T)^{-1}\boldsymbol{C}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1}\right\}(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}). \tag{4.7}$$

Since $\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}$ is proportional to $\tilde{\boldsymbol{D}}$, there exists a scalar c such that

$$(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}} - \eta\tilde{\boldsymbol{D}})^{-1} = c\tilde{\boldsymbol{D}}^{-1}.$$

Formula (4.7) can be simplified to

$$\boldsymbol{\theta} = c\left\{\boldsymbol{I} - \tilde{\boldsymbol{D}}^{-1}\boldsymbol{C}^T(\boldsymbol{C}\tilde{\boldsymbol{D}}^{-1}\boldsymbol{C}^T)^{-1}\boldsymbol{C}\right\}\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}.$$

Due to the symmetry of $\tilde{\boldsymbol{D}}$ and with the definition of $\boldsymbol{C} = \boldsymbol{Q}^T\tilde{\boldsymbol{D}}$ we obtain

$$\boldsymbol{\theta} = c\left\{\boldsymbol{I} - \boldsymbol{Q}(\boldsymbol{Q}^T\tilde{\boldsymbol{D}}\boldsymbol{Q})^{-1}\boldsymbol{Q}^T\tilde{\boldsymbol{D}}\right\}\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}.$$

The scalar c can then be scaled so that the side constraint $\boldsymbol{\theta}^T\tilde{\boldsymbol{D}}\boldsymbol{\theta} = 1$ is fulfilled.

Algorithm for the computation of the initial estimates for $\boldsymbol{\beta}_h$ and $\boldsymbol{\theta}_h$

Input: $h, \boldsymbol{Q}_h, \boldsymbol{X}, \boldsymbol{Y}, \lambda$

(i) Compute $\boldsymbol{D} = \frac{1}{n}\boldsymbol{Y}^T\boldsymbol{Y}$.

(ii) Generate $\boldsymbol{\theta}_*$, a random vector from $N(0,1)$ of length K.

(iii) Compute $\hat{\boldsymbol{\theta}}_h = c\left\{\boldsymbol{I} - \boldsymbol{Q}_h(\boldsymbol{Q}_h^T\boldsymbol{D}\boldsymbol{Q}_h)^{-1}\boldsymbol{Q}_h^T\boldsymbol{D}\right\}\boldsymbol{\theta}_*$, with c so that $\hat{\boldsymbol{\theta}}_h^T\boldsymbol{D}\hat{\boldsymbol{\theta}}_h = 1$.

Apply the following steps twice:

1. For fixed $\hat{\boldsymbol{\theta}}_h$, apply sparse least trimmed squares (sparse LTS) regression (Alfons et al., 2013) to the response $\boldsymbol{Y}\hat{\boldsymbol{\theta}}_h$ and predictors $\boldsymbol{X}$.

   Let $a = 0.5n$ and $\|\boldsymbol{r}\|^2_{1:a} = \sum_{i=1}^{a} r^2_{(i)}$ denote the sum of the a smallest squared elements of the vector $\boldsymbol{r}$. The sparse LTS estimator is a robust version of the Lasso and defined as

   $$\min_{\boldsymbol{\beta}} \frac{1}{a}\|\boldsymbol{Y}\boldsymbol{\theta}_h - \boldsymbol{X}\boldsymbol{\beta}\|^2_{(1):(a)} + \lambda\|\boldsymbol{\beta}\|_1.$$

   As in Alfons et al. (2013), a re-weighting step is carried out afterwards yielding $\hat{\boldsymbol{\beta}}_h$.

2. For fixed $\hat{\boldsymbol{\beta}}_h$, apply least absolute deviation (LAD) regression with response $\boldsymbol{X}\hat{\boldsymbol{\beta}}_h$ and predictor matrix $\boldsymbol{Y}$:

   $$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\boldsymbol{Y}\boldsymbol{\theta} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_h\|_1.$$

The LDA estimator is robust with regard to outliers in the dependent variable, but not to leverage points (i.e. outliers in the covariate space). Since the covariates are dummy variables here, leverage points cannot occur. Then we apply the transformation for satisfying the side constraints:

$$\hat{\boldsymbol{\theta}}_h = c\left\{\boldsymbol{I} - \boldsymbol{Q}_h(\boldsymbol{Q}_h^T\boldsymbol{D}\boldsymbol{Q}_h)^{-1}\boldsymbol{Q}_h^T\boldsymbol{D}\right\}\boldsymbol{\theta}^*.$$

Output: Initial estimators $\hat{\boldsymbol{\beta}}_h$ and $\hat{\boldsymbol{\theta}}_h$

CHAPTER 5

Robust and sparse estimation methods for high-dimensional linear and logistic regression

Abstract: Fully robust versions of the elastic net estimator are introduced for linear and logistic regression. The algorithms used to compute the 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. A final reweighting step improves 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 and by good performance in real data examples.

Key words: Elastic net penalty, Least trimmed squares, C-step algorithm, High-dimensional data, Robustness, Sparse estimation

5.1 Introduction

Let us consider the linear regression model which assumes the linear relationship between the predictors $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the predictand $\mathbf{y} \in \mathbb{R}^{n \times 1}$,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{5.1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ are the regression coefficients and $\boldsymbol{\varepsilon}$ is the error term assumed to have standard normal distribution. For simplicity's sake, we assume that $\mathbf{y} = (y_1, \ldots, y_n)^T$ is centered to mean zero, and the columns of $\mathbf{X}$ are mean-centered and scaled to variance one. The ordinary least squares (OLS) regression estimator is the common choice in situations where n the number of observations in the data set is greater than p the number of predictor variables. However, in presence of multicollinearity among predictors, the OLS estimator becomes unreliable, and if p exceeds n it cannot even be computed. Several alternatives have been proposed in this case; here we focus on the class of shrinkage estimators which penalize the residual sum-of-squares. The ridge estimator uses an l_2 penalty on the regression coefficients Hoerl and Kennard (1970), while the lasso estimator uses an l_1 penalty instead Tibshirani (2011). Although this no longer allows for a closed form solution for the estimated regression coefficients, the lasso estimator becomes *sparse*, which means that some of the regression coefficients are shrunk to zero. This means that lasso acts like a variable selection method by returning a smaller subset of variables relevant for the model. 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 contribute very little information to the model.

There is also 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 might perform poorly. As a way out, the elastic net (*enet*) estimator has been introduced Zou and Hastie (2005), which combines both l_1 and l_2 penalties:

$$\hat{\boldsymbol{\beta}}_{enet} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda P_\alpha(\boldsymbol{\beta}) \right\} \tag{5.2}$$

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

$$P_\alpha(\boldsymbol{\beta}) = (1-\alpha)\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p}\left[(1-\alpha)\frac{1}{2}\beta_j^2 + \alpha|\beta_j|\right]. \tag{5.3}$$

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 shrinks the coefficients according to ridge regression. For an overview of sparse methods, see Filzmoser et al. (2012).

A further limitation of the previously mentioned estimators is their lack of robustness with regard to outliers. In practice, the presence of outliers in data is quite common, and thus robust statistical methods are frequently used, see, for example Liang and Kvalheim (1996); Liang (1996). 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) Maronna et al. (2006). The Least Trimmed Squares (LTS) has been among the first robust estimators proposed which is fully robust against both types of outliers Rousseeuw and Leroy (2003). It is defined as

$$\hat{\boldsymbol{\beta}}_{LTS} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{h} r_{(i)}^2(\boldsymbol{\beta}), \tag{5.4}$$

where the $r_{(i)}$ are the ordered absolute residuals $|r_{(1)}| \leq |r_{(2)}| \leq \cdots \leq |r_{(n)}|$, and $r_i = y_i - \mathbf{x}_i^T\boldsymbol{\beta}$ Rousseeuw (1984). 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 Rousseeuw (1984). The LTS estimator also became popular due to the proposal of a quick algorithm for its computation, the so-called FAST-LTS algorithm Rousseeuw and Van Driessen (2006). 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 OLS estimator can be applied. This only works for $n > p$, but recently the sparse LTS regression estimator has been proposed for high-dimensional problems Alfons et al. (2013):

$$\hat{\boldsymbol{\beta}}_{sparseLTS} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{h} r_{(i)}^2(\boldsymbol{\beta}) + h\lambda\|\boldsymbol{\beta}\|_1 \right\}. \tag{5.5}$$

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 sparse LTS

estimator is robust with regard to both vertical outliers and leverage points, and a fast algorithm has also been developed for its computation Alfons (2013).

The contribution of this work is twofold: A new sparse and robust 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 when there is high multicollinearity among the predictors Tibshirani (2011). As a second contribution, a robust elastic net version of logistic regression is introduced for problems where the response $\mathbf{y}$ is a binary variable encoded with $y_i \in \{0, 1\}$, referring to the class memberships of two groups. The logistic regression model is $y_i = \pi_i + \varepsilon_i$, for $i = 1, \ldots, n$, where π_i denotes the conditional probability for the ith observation,

$$\pi_i = \Pr(y_i = 1|\mathbf{x}_i) = \frac{e^{\mathbf{x}_i^T \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}}}, \tag{5.6}$$

and ε_i is the error term assumed to have binomial distribution. 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,

$$\hat{\boldsymbol{\beta}}_{ML} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} d(\mathbf{x}_i^T \boldsymbol{\beta}, y_i), \tag{5.7}$$

with the deviances

$$d(\mathbf{x}_i^T \boldsymbol{\beta}, y_i) = -y_i \log \pi_i - (1 - y_i) \log(1 - \pi_i) = -y_i \mathbf{x}_i^T \boldsymbol{\beta} + \log\left(1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}}\right). \tag{5.8}$$

The estimation of the model parameters with this method is not reliable when there is multicollinearity among the predictors and is not feasible when $p > n$. To solve these problems, Friedman et al. Friedman et al. (2010) suggested minimizing a penalized negative log-likelihood function,

$$\hat{\boldsymbol{\beta}}_{enet} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} d(\mathbf{x}_i^T \boldsymbol{\beta}, y_i) + n\lambda P_\alpha(\boldsymbol{\beta}) \right\}. \tag{5.9}$$

Here, $P_\alpha(\boldsymbol{\beta})$ is the elastic net penalty as given in Equation (5.3), and thus this estimator extends (5.2) 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 Albert and Anderson (1984); Friedman et al. (2010, 2016). Robustness can be achieved by trimming the penalized log-likelihood function, and using weights as proposed in the context of

robust logistic regression Croux and Haesbroeck (2003); Bianco and Yohai (1996). These weights can also be applied in a reweighting step which increases the efficiency of the robust elastic net logistic regression estimator.

The outline of this paper is as follows: In Section 5.2, we introduce the robust and sparse linear regression estimator and provide a detailed algorithm for its computation. Section 5.3 presents the robust elastic net logistic regression estimator. Some important details which are different from the linear regression algorithm are mentioned here. Section 5.4 explains how the tuning parameters for the proposed estimators can be selected; we prefer an approach based on cross-validation. Since LTS estimators possess a rather low statistical efficiency, a reweighting step is introduced in Section 5.5 to increase the efficiency. The properties of the proposed estimators are investigated in simulation studies in Section 5.6, and Section 5.7 shows the performance using real data examples. Section 5.8 provides some insight into the computation time of the algorithms, and Section 5.9 presents out conclusion.

5.2 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

$$Q(H, \boldsymbol{\beta}) = \sum_{i \in H} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + h\lambda P_\alpha(\boldsymbol{\beta}) \tag{5.10}$$

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 (5.3). We call this estimator the *enet-LTS* estimator, since it uses a trimmed sum of squared residuals, like the sparse LTS estimator (5.5). The minimum of the objective function (5.10) determines the optimal subset of size h,

$$H_{opt} = \underset{H \subseteq 1,2,\ldots,n : |H| = h}{\arg\min} Q(H, \hat{\boldsymbol{\beta}}_H), \tag{5.11}$$

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

$$\hat{\boldsymbol{\beta}}_{enetLTS} = \arg\min Q(H_{opt}, \boldsymbol{\beta}). \tag{5.12}$$

It is not trivial to identify this optimal subset, and in practice 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 h smallest squared residuals

$$r_{k,i}^2 = (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{H_k})^2, \quad \text{for } i = 1, \ldots, n. \tag{5.13}$$

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

$$Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_{k+1}}) \leq Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_k}) \leq Q(H_k, \hat{\boldsymbol{\beta}}_{H_k}) \tag{5.14}$$

This means that the C-steps decrease the objective function (5.10) 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 outlined in Alfons et al. (2013), 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

$$H_{el}^s = \{j_1^s, j_2^s, j_3^s\}, \tag{5.15}$$

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 (5.10) 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 now apply this procedure for a grid of values in the interval $\alpha \in [0, 1]$ and $\lambda \in [0, \lambda_0]$. In practice, this may still be quite time-consuming, and therefore, for a new parameter combination, the best subset of the neighbouring 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.

The choice of the optimal tuning parameters α_{opt} and λ_{opt} is detailed in Section 5.4. The subset corresponding to the optimal tuning parameters is the optimal subset of size

h. The enet-LTS estimator is then calculated based on the optimal subset with α_{opt} and λ_{opt}.

5.3 Robust and sparse logistic regression with elastic net penalty

Based on the definition (5.9) of the elastic net logistic regression estimator, it is straightforward to define the objective function of its robust counterpart based on trimming,

$$Q(H, \boldsymbol{\beta}) = \sum_{i \in H} d(\mathbf{x}_i^T \boldsymbol{\beta}, y_i) + h\lambda P_\alpha(\boldsymbol{\beta}), \tag{5.16}$$

where again $H \subseteq \{1, 2, \ldots, n\}$ with $|H| = h$, and P_α is the elastic net penalty as defined in Equation (5.3). As outlined in Section 5.2, 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 what follows, 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 (5.13). Now the h-subsets are determined according to the indexes of those observations where the deviances $d(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{H_k}, y_i)$ have the smallest values. However, we must ensure that the selected observations have the same group proportions as the original data. 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 (5.15). Now four observations are randomly selected to form the elemental subsets, two from each group. This makes it possible 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 Croux and Haesbroeck (2003) in the context of robust

logistic regression (but not in high dimensions). These authors replace the deviance function (5.8) used in (5.7) by a function φ_{BY} to define the Bianco Yohai (BY) estimator

$$\hat{\boldsymbol{\beta}}_{BY} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \varphi(\mathbf{x}_i^T \boldsymbol{\beta}; y_i), \tag{5.17}$$

a highly robust logistic regression estimator, see also Bianco and Yohai (1996). The form of the function φ_{BY} is shown in Figure 5.1, see Croux and Haesbroeck (2003) 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. We observe the opposite behaviour for the scores of group 0, see Figure 5.1. When evaluating an h-subset, the sum of 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 $\varphi_{BY}(\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_H)$ for all $i \in H$ forms the best index set.

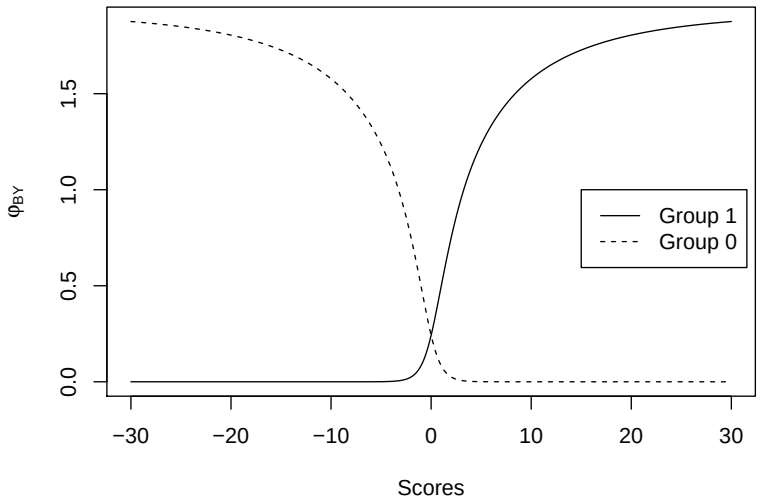


Figure 5.1: Function φ_{BY} used for evaluating an h-subset, based on the scores $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}$ for the two groups.

The selection of the optimal parameters α_{opt} and λ_{opt} is discussed in Section 5.4. 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}.

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. While carrying out 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 Alfons et al. (2013). The same procedure is applied for logistic regression, except for centering the predictand. In the end, the coefficients are back-transformed to the original scale.

5.4 Selection of the tuning parameters

Sections 5.2 and 5.3 outlined the algorithms to arrive at a best subset for robust elastic net linear and 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 linear regression we take the root mean squared prediction error (RMSPE)

$$\text{RMSPE}(\alpha, \lambda) = \sqrt{\frac{1}{h} \sum_{i=1}^{h} r_i^2(\hat{\boldsymbol{\beta}}_{\alpha,\lambda})} \tag{5.18}$$

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

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

$$\text{MNLL}(\alpha, \lambda) = \frac{1}{h} \sum_{i=1}^{h} d_i(\hat{\boldsymbol{\beta}}_{\alpha,\lambda}), \tag{5.19}$$

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

Note that the evaluation criteria given by (5.18) and (5.19) are robust with regard to outliers, because they are 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 calculate 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. Alfons et al. (2013) which is based on the Pearson correlation between y and the jth predictor variable x_j on winsorized data. For logistic regression, we replaced the Pearson correlation by a robustified point-biserial correlation: denote by n_0 and n_1 the group sizes of the two groups, 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 y and x_j is defined as

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

where $\text{MAD}(x_j)$ is the MAD of x_j, and $n = n_0 + n_1$.

5.5 Reweighting step

The LTS estimator has a low efficiency, and thus it is common to use a reweighting step Rousseeuw and Leroy (2003). 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. Alfons et al. (2013), which is based on

standardized residuals. In case of logistic regression we compute the Pearson residuals which are approximately standard normally distributed and given by

$$r_i^s = \frac{y_i - \pi_i}{\pi_i (1 - \pi_i)}, \tag{5.20}$$

with π_i the conditional probabilities from (5.6).

For simplicity's sake, let us also denote the standardized residuals from the linear regression case by r_i^s. Then the weights are defined by

$$w_i = \begin{cases} 1, & \text{if } |r_i^s| \leq \Phi^{-1}(1-\delta) \\ 0, & \text{if } |r_i^s| > \Phi^{-1}(1-\delta) \end{cases} \quad i = 1, 2, \ldots, n, \tag{5.21}$$

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

$$\hat{\boldsymbol{\beta}}_{reweighted} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} w_i f(\mathbf{x}_i; y_i) + \lambda_{upd} n_w P_{\alpha_{opt}}(\boldsymbol{\beta}) \right\}, \tag{5.22}$$

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 from h observations, the penalty can act slightly differently in (5.22) 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.

5.6 Simulation studies

In this section, the performance of the new estimators is compared with different sparse estimators in different scenarios. We consider both the raw and the reweighted versions of the enet-LTS estimators, and therefore aim to show how the reweighting step improves the methods. The raw and reweighted enet-LTS estimators are compared with their classical, non-robust counterparts, which are the linear and logistic regression estimators with elastic net penalty Friedman et al. (2010). In case of linear regression we also compare with the reweighted sparse LTS estimator of Alfons et al. (2013). All robust estimators are calculated taking the subset size $h = \lfloor (n+1) \cdot 0.75 \rfloor$ such that their performances are directly comparable.

For each replication, we choose the optimal tuning parameters α_{opt} and λ_{opt} over the grids α and λ with 5-times repeated 5-fold CV as described in Section 5.4. To select the tuning parameters for the classical estimators with elastic net penalty, we first draw the same grid for α, namely $\alpha \in [0, 1]$, with 41 equally spaced grid points. Then we use 5-fold CV as provided by the R package *glmnet*, which automatically checks the model quality for a sequence of values for λ, taking the mean squared error as an evaluation criterion. Finally, the tuning parameters corresponding to the smallest value of the minimum cross-validated error are determined as the optimal tuning parameters. In order to be coherent with our evaluation, the tuning parameters for the sparse LTS estimator are determined in the same way as for the enet-LTS estimator. All simulations are carried out in R R Development Core Team (2017).

Note that we simulated the data sets with intercept. As described at the end of Section 5.3, the data are centered and scaled at the beginning of the algorithm and only in the final step are the coefficients back-transformed to the original scale, where the estimate of the intercept is computed.

***Sampling schemes for 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 application, 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 a value of one. The response variable is calculated by

$$y_i = 1 + \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \tag{5.23}$$

where the error term ε_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 with 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.

***Sampling schemes for 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

$$y_i = \begin{cases} 0, & \text{if } 1 + \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i \leq 0 \\ 1, & \text{if } 1 + \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i > 0 \end{cases} \quad i = 1, 2, \ldots, n. \tag{5.24}$$

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

Contamination is introduced by adding outliers to the informative variables only. Denote n_0 the number of observations in class 0. Then the first $\lfloor 0.1 n_0 \rfloor$ 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.

***Performance measures*:** To evaluate the different estimators, training and test data sets are generated according to the sampling schemes explained earlier. 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,

$$\text{RMSPE}(\hat{\boldsymbol{\beta}}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}\right)^2}, \tag{5.25}$$

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

$$\text{MNLL}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} d(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\boldsymbol{\beta}}, y_i), \tag{5.26}$$

where y_i and $\mathbf{x}_i$, $i = 1, \ldots, n$, indicate the observations comprising 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

$$\text{MCR} = \frac{m}{n} \tag{5.27}$$

where m is the number of misclassified observations in the test data after fitting the model to the training data. Furthermore, we consider the precision of the coefficient estimate as a quality criterion, defined by

$$\text{PRECISION}(\hat{\boldsymbol{\beta}}) = \sqrt{\sum_{i=0}^{p}\left(\beta_i - \hat{\beta}_i\right)^2}, \tag{5.28}$$

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

$$\text{FPR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j = 0, \ldots, p : \hat{\beta}_j \neq 0 \wedge \beta_j = 0\}|}{|\{j = 0, \ldots, p : \beta_j = 0\}|}, \tag{5.29}$$

$$\text{FNR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j = 0, \ldots, p : \hat{\beta}_j = 0 \wedge \beta_j \neq 0\}|}{|\{j = 0, \ldots, p : \beta_j \neq 0\}|}. \tag{5.30}$$

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 in boxplots. The smaller the value for these criteria, the better the performance of the method.

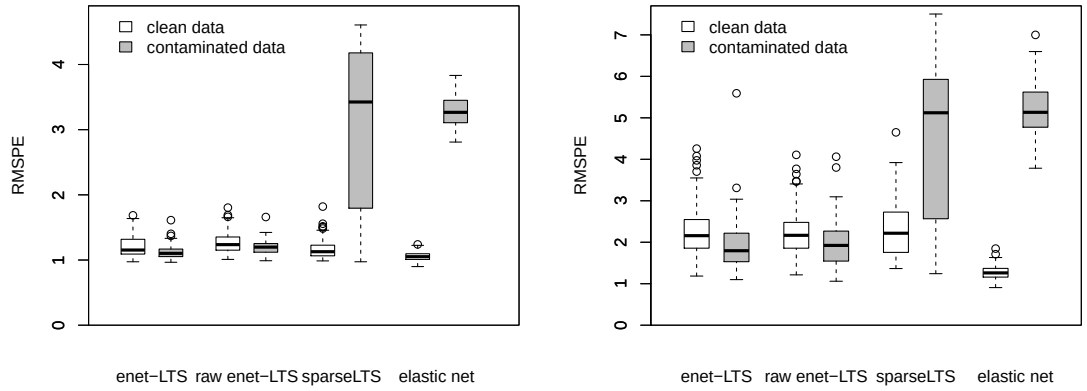


Figure 5.2: Root mean squared prediction error (RMSPE) for linear regression. Left: low-dimensional data set ($n = 150$ and $p = 60$); right: high-dimensional data set ($n = 50$ and $p = 100$).

***Results for linear regression*:** The outcome of the simulations for linear regression is summarized in Figures 5.2–5.5. The left plots in these figures are for the simulations with low-dimensional data, and the right plots for the high-dimensional configuration. Figure 5.2 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 5.3 shows essentially the same behavior. The FPR in Figure 5.4, 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 5.5 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.

***Results for logistic regression*:** Figures 5.6–5.10 summarize the simulation results for logistic regression. As before, the left-handed plots refer to the low-dimensional data, and the right plots to the high-dimensional data. Within one plot, the results

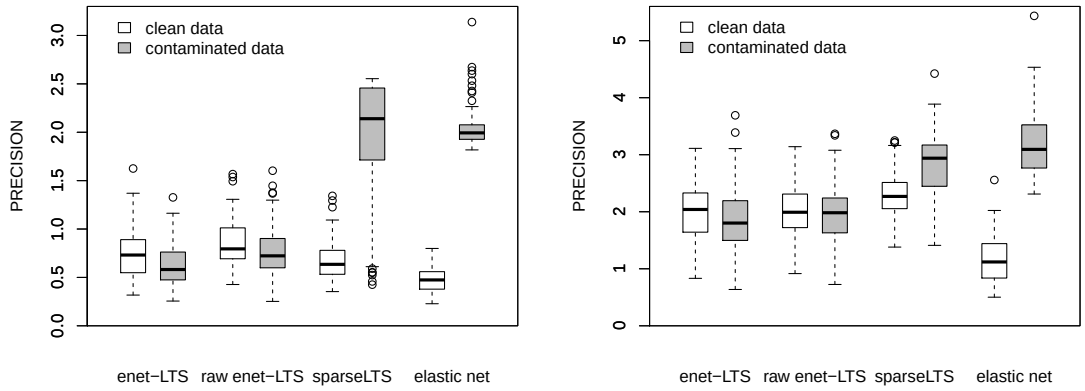


Figure 5.3: Precision of the estimators (PRECISION) for linear regression. Left: low-dimensional data set ($n = 150$ and $p = 60$); right: high-dimensional data set ($n = 50$ and $p = 100$).

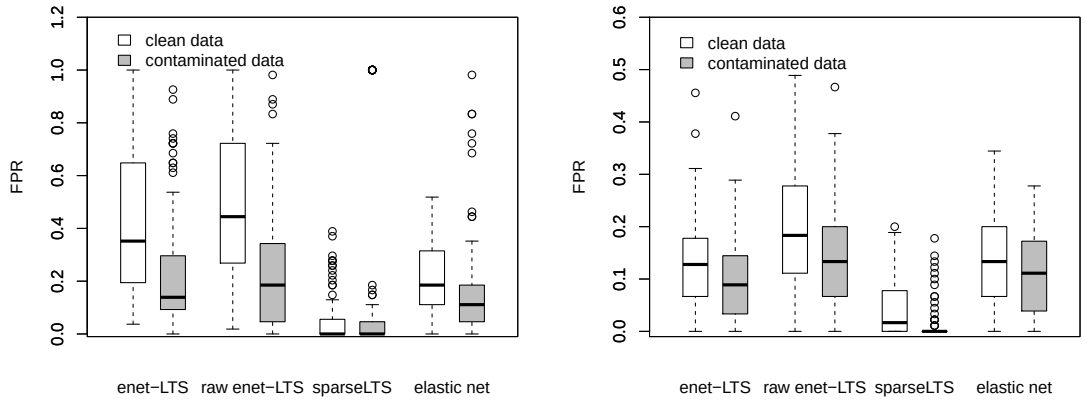


Figure 5.4: False positive rate (FPR) for linear regression. Left: low-dimensional data set ($n = 150$ and $p = 60$); right: high-dimensional data set ($n = 50$ and $p = 100$).

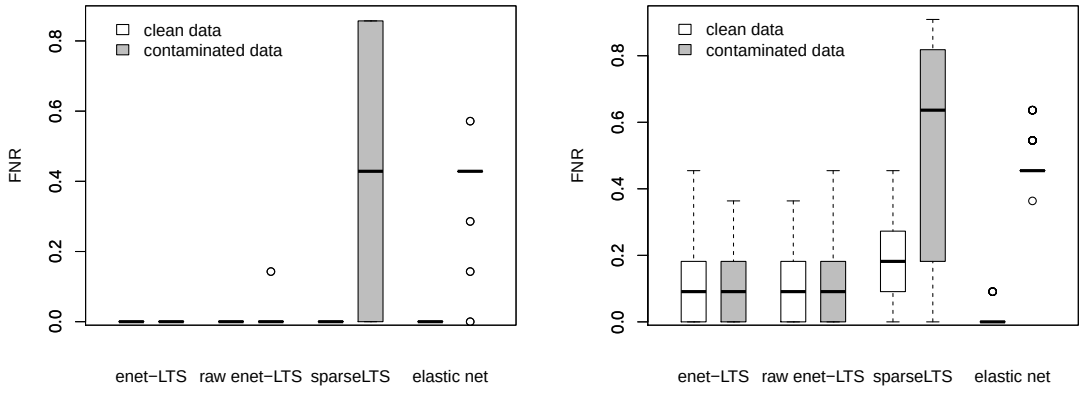


Figure 5.5: False negative rate (FNR) for linear regression. Left: low-dimensional data set ($n = 150$ and $p = 60$); right: high-dimensional data set ($n = 50$ and $p = 100$).

for uncontaminated and contaminated data are compared directly. The misclassification rate in Figure 5.6 is around 10% for all methods, and it is slightly higher in the high-dimensional situation. When there is contamination, however, this rate increases enormously for the classical method elastic net.

The average deviances in Figure 5.7 show that the reweighting of the enet-LTS estimator clearly improves the raw estimate in both low-dimensional and high-dimensional settings. It can also be seen that elastic net is sensitive to outliers. The precision of the parameter estimates in Figure 5.8 reveal a remarkable improvement for the reweighted enet-LTS estimator compared to the raw version, while the contamination does not have any clear effect on the classical elastic net estimator.

The FPR in Figure 5.9 shows a certain difference between uncontaminated and contaminated data for the elastic net, but otherwise the results are quite comparable. A different picture can be seen in the FNR in Figure 5.10, 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 when there is contamination, even though this was not immediatly apparent when we looked at the precision alone, and it also yields reasonable results for clean data.

The results for various different choices of n and p are presented in Table 5.1 and 5.2. Shown are the average values of the quality measures over $m = 100$ replications for clean and contaminated data scenarios, respectively. Table 5.1 and 5.2 support the

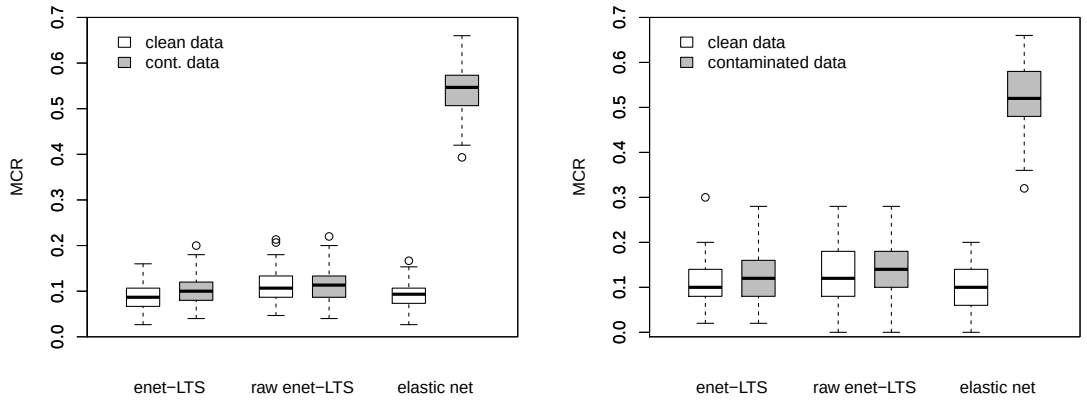


Figure 5.6: Misclassification rate for logistic regression. Left: low-dimensional data set ($n = 150$ and $p = 50$); right: high-dimensional data set ($n = 50$ and $p = 100$).

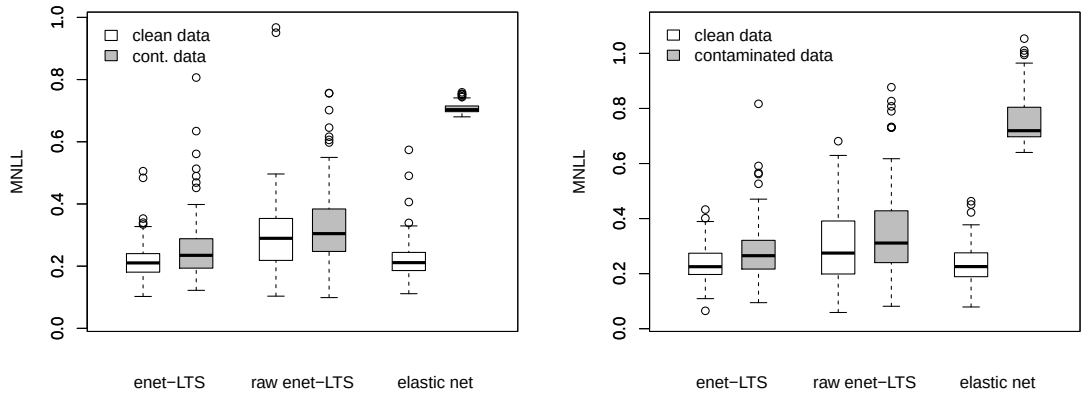


Figure 5.7: The mean of negative likelihood (MNLL) function for logistic regression. Left: low-dimensional data set ($n = 150$ and $p = 50$); right: high-dimensional data set ($n = 50$ and $p = 100$).

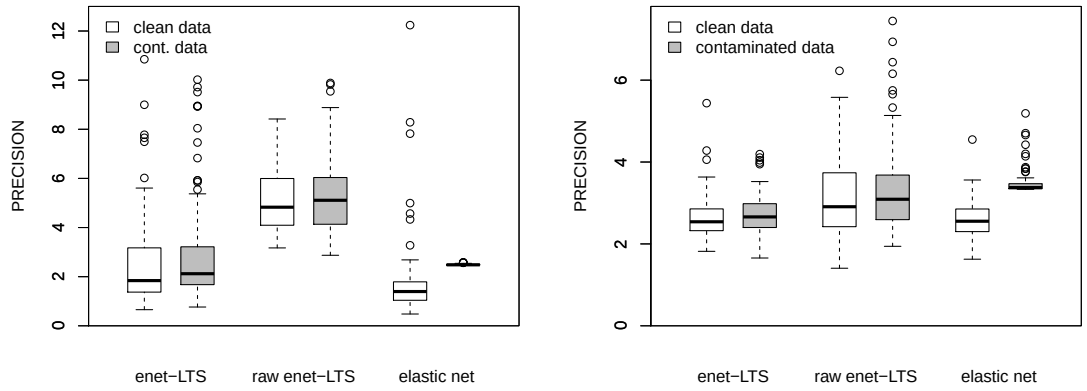


Figure 5.8: Precision of the estimators (PRECISION) for logistic regression. Left: low-dimensional data set ($n = 150$ and $p = 50$); right: high-dimensional data set ($n = 50$ and $p = 100$).

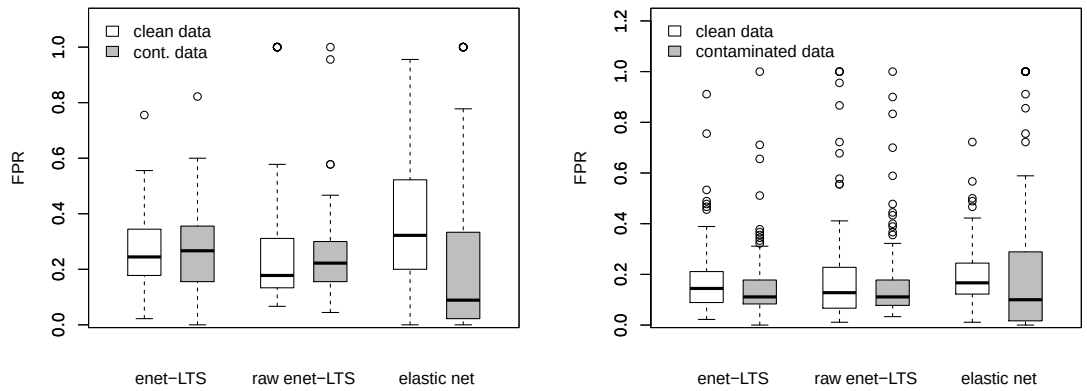


Figure 5.9: False positive rate (FPR) for logistic regression. Left: low-dimensional data set ($n = 150$ and $p = 50$); right: high-dimensional data set ($n = 50$ and $p = 100$).

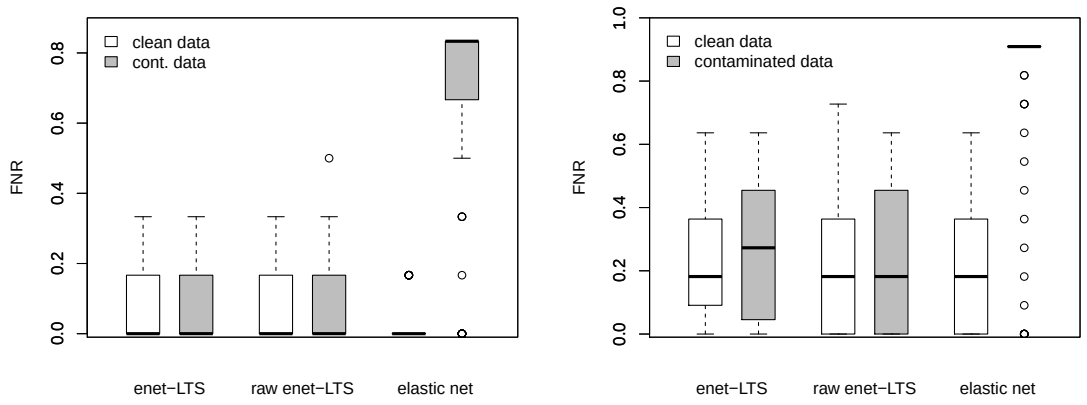


Figure 5.10: False negative rate (FNR) for logistic regression. Left: low-dimensional data set ($n = 150$ and $p = 50$); right: high-dimensional data set ($n = 50$ and $p = 100$).

results given in Figures 5.6–5.10.

5.7 Real data applications

In this section, we will 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 5.4. Model evaluation is done with leave-one-out cross validation, i.e. each observation is used as test observation once, a model is estimated based 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 prevent potential outliers from influencing the evaluation of a model, the 25% trimmed mean of the negative log-likelihoods is calculated in order to compare the methods.

Analysis of meteorite data

The time-of-flight secondary iron mass spectroscope COSIMA Kissel et al. (2007) 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 Schulz et al. (2015). As reference measurements, samples of meteorites provided by the Natural

Table 5.1: Results for logistic regression with no contamination: mean of negative log-likelihood (MNLL), misclassification rate (MCR), bias of the estimators (Bias), false positive rate (FPR) and false negative rate (FNR), averaged over $m = 100$ runs.

		No Contamination				
Setting	Method	MNLL	MCR	Bias	FPR	FNR
with n=25	enet-LTS	0.587	0.262	9.88	0.316	0.494
and p=50	elastic net	0.551	0.268	9.84	0.183	0.514
with n=25	enet-LTS	0.363	0.161	2.95	0.188	0.308
and p=100	elastic net	0.335	0.150	2.80	0.195	0.221
with n=25	enet-LTS	0.587	0.262	9.88	0.316	0.494
and p=1000	elastic net	0.551	0.268	9.84	0.183	0.514
with n=50	enet-LTS	0.278	0.116	2.44	0.224	0.145
and p=50	elastic net	0.279	0.113	2.16	0.282	0.093
with n=50	enet-LTS	0.239	0.103	2.65	0.180	0.224
and p=100	elastic net	0.236	0.102	2.61	0.201	0.195
with n=50	enet-LTS	0.463	0.200	9.73	0.232	0.486
and p=1000	elastic net	0.400	0.177	9.66	0.210	0.334
with n=150	enet-LTS	0.235	0.096	3.00	0.316	0.055
and p=50	elastic net	0.210	0.088	1.61	0.311	0.025
with n=150	enet-LTS	0.183	0.073	2.69	0.180	0.157
and p=100	elastic net	0.182	0.073	2.39	0.309	0.061
with n=150	enet-LTS	0.311	0.136	9.38	0.059	0.590
and p=1000	elastic net	0.257	0.112	9.29	0.180	0.239
with n=500	enet-LTS	0.216	0.097	1.45	0.484	0.000
and p=50	elastic net	0.176	0.075	1.13	0.338	0.002
with n=500	enet-LTS	0.139	0.060	1.92	0.367	0.016
and p=100	elastic net	0.121	0.048	1.57	0.292	0.013
with n=500	enet-LTS	0.182	0.079	8.89	0.075	0.461
and p=1000	elastic net	0.156	0.061	8.83	0.155	0.180

Table 5.2: Results for logistic regression with contamination: mean of negative log-likelihood (MNLL), misclassification rate (MCR), bias of the estimators (Bias), false positive rate (FPR) and false negative rate (FNR), averaged over $m = 100$ runs.

		Contaminated				
Setting	Method	MNLL	MCR	Bias	FPR	FNR
with n=25	enet-LTS	0.216	0.097	1.45	0.484	0.000
and p=50	elastic net	0.176	0.075	1.13	0.338	0.002
with n=25	enet-LTS	0.434	0.190	3.04	0.288	0.271
and p=100	elastic net	0.831	0.518	3.51	0.182	0.789
with n=25	enet-LTS	0.726	0.412	10.04	0.401	0.551
and p=1000	elastic net	0.791	0.497	10.11	0.146	0.880
with n=50	enet-LTS	0.323	0.137	2.27	0.217	0.157
and p=50	elastic net	0.805	0.527	2.71	0.273	0.703
with n=50	enet-LTS	0.253	0.108	2.78	0.189	0.231
and p=100	elastic net	0.776	0.523	3.46	0.194	0.816
with n=50	enet-LTS	0.607	0.322	9.88	0.219	0.605
and p=1000	elastic net	0.756	0.511	10.09	0.079	0.937
with n=150	enet-LTS	0.259	0.102	2.96	0.272	0.060
and p=50	elastic net	0.709	0.543	2.49	0.234	0.682
with n=150	enet-LTS	0.201	0.080	2.79	0.187	0.141
and p=100	elastic net	0.711	0.524	3.36	0.181	0.800
with n=150	enet-LTS	0.409	0.191	9.51	0.077	0.605
and p=1000	elastic net	0.714	0.517	10.07	0.370	0.622
with n=500	enet-LTS	0.216	0.095	1.64	0.462	0.005
and p=50	elastic net	0.701	0.547	2.47	0.308	0.530
with n=500	enet-LTS	0.131	0.056	2.46	0.288	0.034
and p=100	elastic net	0.701	0.534	3.33	0.252	0.664
for n=500	enet-LTS	0.262	0.105	9.05	0.076	0.477
and p=1000	elastic net	0.702	0.526	10.06	0.121	0.807

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 Varmuza et al. (2011) and variables with positive median absolute deviation. So we obtain $p = 1540$ variables. Furthermore, the data is normalized to have constant row sum 100.

Table 5.3 summarizes the results of 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.

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

Table 5.3: 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.

Figure 5.11 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 and they are marked as horizontal lines in Figure 5.11. Closer investigation showed that these spectra lie on the outer border of the measurement area and might have been 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 5.12, where we can see that in some variables they have particularly large values compared to the majority of the group.

Analysis of the glass vessels data

Archaeological glass vessels were 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

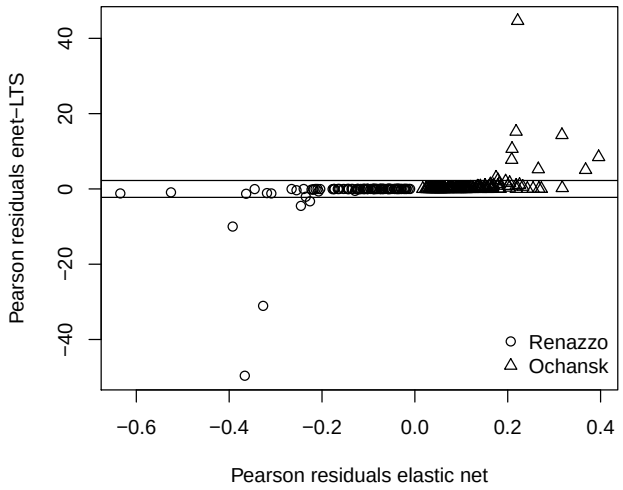


Figure 5.11: Renazzo and Ochansk: the Pearson residuals of elastic net and the raw enet-LTS estimator. The horizontal lines indicate the 0.0125 and the 0.9875 quantiles of the standard normal distribution.

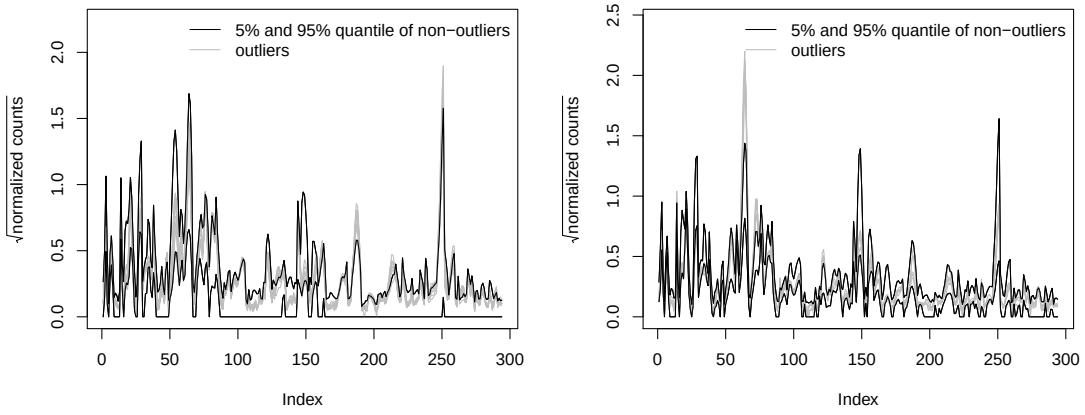


Figure 5.12: The index refers to the index of the variables included in the model of raw enet-LTS. The detected outliers are visualized by grey lines, while the black lines represent the 5% and 95% quantile of the non-outlying spectra for Ochansk (left) and Renazzo (right).

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

Table 5.4: 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.

Janssens et al. (1998). Four different groups were identified, i.e. sodic, potassic, potasso-calcic and calcic glass vessels. To demonstrate 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 is the potasso-calcic group with 10 spectra. As in Filzmoser et al. (2008), we remove variables with MAD equal to zero, resulting in $p = 1905$ variables.

The quality of the selected models is described in Table 5.4. 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 $\alpha_{opt} = 0.05$.

We can expect the coefficient estimates to behave differently. Figure 5.13 (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. Furthermore, 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 5.13 (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.

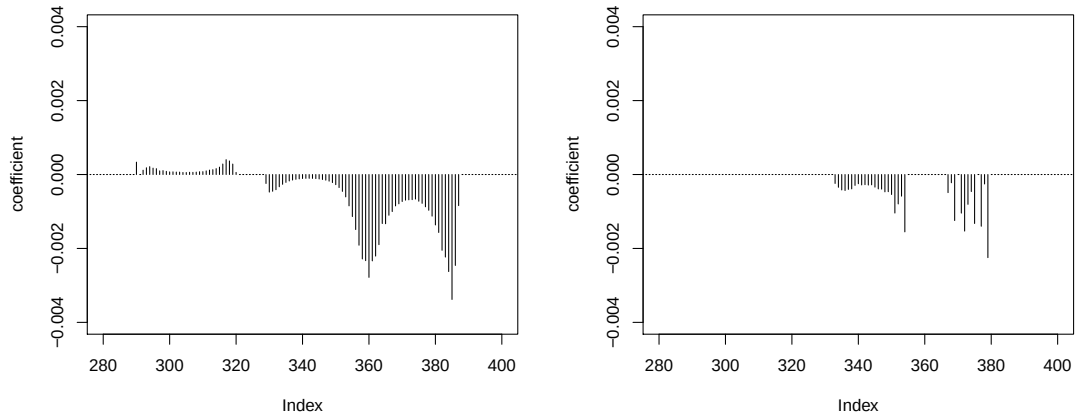


Figure 5.13: Glass vessels: coefficient estimate of the reweighted enet-LTS model (left) and coefficient estimate of the elastic net mode (right) for a selected variable range.

5.8 Computation time

For our algorithm, we employ the classical elastic net estimator as it is implemented in the R package *glmnet* Friedman et al. (2016). So, it is natural to compare the computation time of our algorithm with this method. In the linear regression case we also make a comparison with the sparse LTS estimator implemented in the R package *robustHD* Alfons (2013). To calculate the estimators, we take a grid of five values for both tuning parameters α and λ. The data sets are simulated as in Section 5.6 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 5.14 (left: linear regression, right: logistic regression), the CPU time is reported in seconds, as an average over 5 repetitions. 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 5.15. The computations are performed on an Intel Core 2 Q9650 @ 3000 GHz×4 processor.

Let us first consider the relationship of the computation time to the number of variables p for linear regression, shown in the left plot of Figure 5.14. Sparse LTS increases greatly with the number of variables p since it is based on the LARS algorithm which has a computational complexity of $\mathcal{O}(p^3 + np^2)$ Efron et al. (2004). Even 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

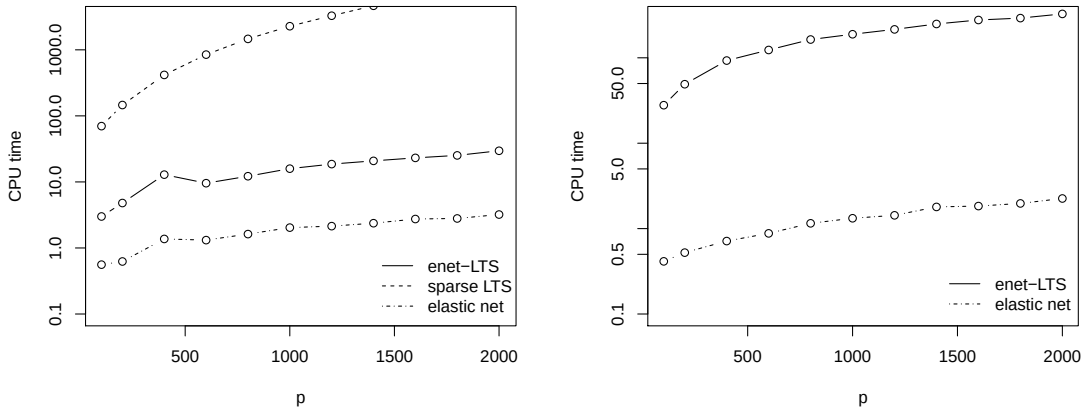


Figure 5.14: CPU time in seconds (log-scale), averaged over 5 repetitions, for fixed $n = 150$ and varying p; left: for linear regression; right: for logistic regression.

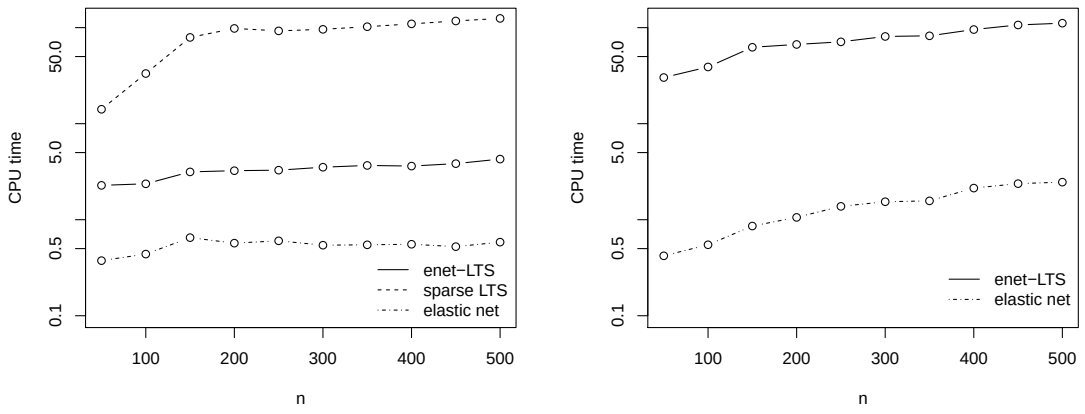


Figure 5.15: CPU time in seconds (log-scale), averaged over 5 repetitions, for fixed $p = 100$ and varying n; left: for linear regression; right: for logistic regression.

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 value of the objective function, further C-steps are performed. In contrast, the enet-LTS estimator starts with 500 elemental subsets for only 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. A total of approximately 1,700 elastic net models are estimated in this procedure. This number is 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 5.15 (left).

The results for the computation time in logistic regression are presented in Figure 5.14 (right) and 5.15 (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.

5.9 Conclusions

In this paper, robust methods for linear and logistic regression using the elastic net penalty were introduced. This penalty allows for variable selection, can deal with high multicollinearity among the variables, and is thus very appropriate in high-dimensional sparse settings. Robustness has been achieved by using trimming. This 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, and also with respect to computation time. Particularly, it was shown that the idea of using "warm starts" for parameter tuning saves computation time, while still maintaining precision. A competing method for robust high-dimensional linear regression, the sparse LTS estimator Alfons (2013), does not use this idea, and is thus much less attractive concerning computation time, especially when there are 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 highly relevant, since in many modern applications in chemometrics or bio-informatics, one is confronted with data information from two groups, and one has to find a classification rule and to identify marker variables which support the rule. 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 the presence of outliers the new proposal still works well, while its classical non-robust counterpart can lead to poor performance.

Identifying outliers beforehand and continuing with the classical procedures on the cleaned data would be an alternative to using robust estimators. However, it is not at all trivial to identify multivariate outliers in high dimensions, and there are almost no methods available for this purpose (see e.g. Filzmoser et al. (2008)). Besides, we are mainly interested in model outliers. Our model is sparse which means that some variables are not included and therefore outliers in these variables do not need to be down-weighted.

Note that in Park and Konishi (2016), a logistic regression method with elastic net penalty is proposed using weights to reduce the influence of outliers. Their approach is to perform outlier detection in a 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 likelihoods 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.

The algorithms used to compute the proposed estimators are implemented in R functions, which are available upon request from the authors. The basis for the computation of the robust estimator is the R package *glmnet* Friedman et al. (2016). This package also implements the case of multinomial and Poisson regression. Naturally, a further extension of the algorithms introduced here could explore these areas of research. Further work will be devoted to the theoretical properties of the family of enet-LTS estimators.

Acknowledgments

This work is supported in part by the Austrian Science Fund (FWF), project P 26871-N20 and by grant TUBITAK 2214/A from the Scientific and Technological Research Council of Turkey (TUBITAK).

The authors thank F. Brandstätter, L. Ferrière, and C. Koeberl (Natural History Museum Vienna, Austria) for providing meteorite samples, C. Engrand (Centre de Sciences Nucléaires et de Sciences de la Matière, Orsay, France) for sample preparation, and M. Hilchenbach (Max Planck Institute for Solar System Research, Göttingen, Germany) for TOF-SIMS measurements. The authors are grateful to Kurt Varmuza for valuable feedback on the results of the meteorite data.

List of Figures

List of Tables

Bibliography

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

Alfons, A. (2013). robustHD: Robust methods for high dimensional data. R Foundation for Statistical Computing, Vienna, Austria. R package version 0.4.0.

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

Allen, G., Peterson, C., Vannucci, M., and Maletić-Savatić, M. (2013). Regularized partial least squares with an application to nmr spectroscopy. Statistical Analysis and Data Mining, 6:302–314.

Armanino, C., Leardi, R., Lanteri, S., and Modi, G. (1989). Chemometric analysis of tuscan olive oils. Chemometrics and Intelligent Laboratory Systems, 5(4):343–354.

Barker, M. and Rayens, W. (2003). Partial least squares for discrimination. Journal of Chemometrics, 17(3):166–173.

Barker, R. (2000). Partial least squares for discrimination. PhD thesis, University of Kentucky.

Bennett, K. P., Fayyad, U., and Geiger, D. (1999). Density-based indexing for approximate nearest-neighbor queries. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 233–243. ACM.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In International conference on database theory, pages 217–235. Springer.

Bianco, A. M. and Yohai, V. J. (1996). Robust estimation in the logistic regression model. In Robust statistics, data analysis, and computer intensive methods, pages 17–34. Springer.

Chiang, L., Russell, E., and Braatz, R. (2000). Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. Chemometrics and Intelligent Laboratory Systems, 50(2):243–252.

Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(1):3–25.

Chung, D. and Keleş, S. (2010). Sparse partial least squares classification for high dimensional data. Statistical Applications in Genetics and Molecular Biology, 9(1).

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2012). Sparse discriminant analysis. Technometrics, 53(4):406–413.

Clemmensen, L. and Kuhn, M. (2012). sparseLDA: Sparse Discriminant Analysis. R package version 0.1-6.

Croux, C., Filzmoser, P., and Fritz, H. (2013). Robust sparse principal component analysis. Technometrics, 55(2):202–214.

Croux, C. and Haesbroeck, G. (2003). Implementing the Bianco and Yohai estimator for logistic regression. Computational statistics & data analysis, 44(1):273–295.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). Pattern classification. John Wiley & Sons.

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

Filzmoser, P., Gschwandtner, 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.

Filzmoser, P., Maronna, R., and Werner, M. (2008). Outlier identification in high dimensions. Computational Statistics & Data Analysis, 52(3):1694–1711.

Filzmoser, P., Serneels, S., Maronna, R., and Van Espen, P. (2009). Robust multivariate methods in chemometrics. In Brown, S., Tauler, R., and Walczak, B., editors, Comprehensive Chemometrics, volume 3, pages 681–722. Elsevier, Oxford.

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. R package version 2.0-5.

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.

Hall, P., Marron, J., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(3):427–444.

Hampel, F. (1974). The influence curve and its role in robust estimation. Journal of the American Statistical Association, 69(346):383–393.

Hampel, F., Ronchetti, E., Rousseeuw, P., and Stahel, W. (1986). Robust Statistics: The Approach Based on Influence Functions. Wiley.

Hastie, T., Tibshirani, R., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. Journal of the American statistical association, 89(428):1255–1270.

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

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

Hoffmann, I., Filzmoser, P., Serneels, S., and Varmuza, K. (2016). Sparse and robust PLS for binary classification. Journal of Chemometrics, 30(4):153–162.

Hoffmann, I., Serneels, S., Filzmoser, P., and Croux, C. (2015). Sparse partial robust M regression. Chemometrics and Intelligent Laboratory Systems, 149:50 – 59.

Hubert, M., Rousseeuw, P., and Van Aelst, S. (2008). High-breakdown robust multivariate methods. Statistical Science, 23(1):92–119.

Hubert, M. and Van Driessen, K. (2004). Fast and robust discriminant analysis. Computational Statistics & Data Analysis, 45(2):301–320.

Jackson, D. (1993). Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. Ecology, pages 2204–2214.

Janssens, K., Deraedt, I., Freddy, A., and Veekman, J. (1998). Composition of 15-17 th century archeological glass vessels excavated in Antwerp, Belgium. Mikrochima Acta, 15:253–267.

Johnson, R., Wichern, D., et al. (2002). Applied multivariate statistical analysis, volume 5 (8). Prentice hall Upper Saddle River, NJ.

Kemsley, E. (1996). Discriminant analysis of high-dimensional data: a comparison of principal components analysis and partial least squares data reduction methods. Chemometrics and Intelligent Laboratory Systems, 33(1):47–61.

Kettaneh, N., Berglund, A., and Wold, S. (2005). PCA and PLS with very large data sets. Computational Statistics & Data Analysis, 48(1):69–85.

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

Lê Cao, K., Rossouw, D., Robert-Granié, C., and Besse, P. (2008). A sparse PLS for variable selection when integrating omics data. Statistical Applications in Genetics and Molecular Biology, 7:35.

Lee, D., Lee, W., Lee, Y., and Pawitan, Y. (2011). Sparse partial least-squares regression and its applications to high-throughput data analysis. Chemometrics and Intelligent Laboratory Systems, 109(1):1–8.

Liang, Y.Z. anf Fang, K. (1996). Robust multivariate calibration algorithm based on least median of squares and sequential number theory optimization method. Analyst, 121(8):1025–1029.

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

Liebmann, B., Filzmoser, P., and Varmuza, K. (2010). Robust and classical PLS regression compared. Journal of Chemometrics, 24:111–120.

M. Salibián-Barrera, S. V. A. and Willems, G. (2006). Principal components analysis based on multivariate MM-estimators with fast and robust bootstrap. Journal of the American Statistical Association, 101:1198–1211.

Maronna, R., Martin, R., and Yohai, V. (2006). Robust Statistics: Theory and Methods. Wiley, New York.

McLachlan, G. (2004). Discriminant analysis and statistical pattern recognition, volume 544. John Wiley & Sons.

Nguyen, D. and Rocke, D. (2002). Tumor classification by partial least squares using microarray gene expression data. Bioinformatics, 18(1):39–50.

Öllerer, V. (2015). Robust and sparse estimation in high-dimensions. PhD thesis, KU Leuven.

Oshima, R. G., Baribault, H., and Caulín, C. (1996). Oncogenic regulation and function of keratins 8 and 18. Cancer and Metastasis Reviews, 15(4):445–471.

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

Pérez-Enciso, M. and Tenenhaus, M. (2003). Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach. Human Genetics, 112(5-6):581–592.

R Core Team (2016). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

R Development Core Team (2017). R: A Language and Environment for Statistical Computing, Vienna, Austria. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rossini, K., Verdun, S., Cariou, V., Qannari, E., and Fogliatto, F. (2012). PLS discriminant analysis applied to conventional sensory profiling data. Food Quality and Preference, 23(1):18–24.

Rousseeuw, P. (1984). Least median of squares regression. Journal of the American Statistical Association, 79(388):871–880.

Rousseeuw, P. (1985). Multivariate estimation with high breakdown point. Mathematical statistics and applications, 8:283–297.

Rousseeuw, P. and Croux, C. (1993). Alternatives to the median absolute deviation. Journal of the American Statistical Association, 88:1273–1283.

Rousseeuw, P. and Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. Technometrics, 41(3):212–223.

Rousseeuw, P. and Leroy, A. (2003). Robust Regression and Outlier Detection. Wiley 2nd edition: John Wiley & Sons, New York.

Rousseeuw, P. and Van Driessen, K. (2006). Computing LTS regression for large data sets. Data mining and knowledge discovery, 12(1):29–45.

Sääksjärvi, E., Khalighi, M., and Minkkinen, P. (1989). Waste water pollution modelling in the southern area of Lake Saimaa, Finland, by the SIMCA pattern recognition method. Chemometrics and Intelligent Laboratory Systems, 7(1):171–180.

Schelfhout, L. J., Van Muijen, G. N., and Fleuren, G. J. (1989). Expression of keratin 19 distinguishes papillary thyroid carcinoma from follicular carcinomas and follicular thyroid adenoma. American Journal of Clinical Pathology, 92(5):654–658.

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

Serneels, S., Croux, C., Filzmoser, P., and Espen, P. V. (2005). Partial robust M-regression. Chemometrics and Intelligent Laboratory Systems, 79(1-2):55 – 64.

Serneels, S. and Hoffmann, I. (2014). sprm: Sparse and Non-sparse Partial Robust M Regression. R package version 1.0.

Serneels, S. and Hoffmann, I. (2015). sprm: Sparse and Non-Sparse Partial Robust M Regression and Classification. R package version 1.2.

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

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society, B, 73(3):273–282.

Todorov, V. (2016). rrcovHD: Robust Multivariate Methods for High Dimensional Data. R package version 0.2-4.

Todorov, V. and Pires, A. (2007). Comparative performance of several robust linear discriminant analysis methods. REVSTAT Statistical Journal, 5:63–83.

Vanden Branden, K. and Hubert, M. (2005). Robust classification in high dimensions based on the SIMCA method. Chemometrics and Intelligent Laboratory Systems, 79(1):10–21.

Varmuza, K., Engrand, C., Filzmoser, P., Hilchenbach, M., Kissel, J., Krüger, H., Silén, J., and Trieloff, M. (2011). Random projection for dimensionality reduction – applied to time-of-flight secondary ion mass spectrometry data. Analytica Chimica Acta, 705(1):48–55.

Witten, D. and Tibshirani, R. (2011). Penalized classification using Fisher's linear discriminant. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(5):753–772.

Wold, H. (1965). Multivariate analysis. In Krishnaiah, P., editor, Proceedings of an International Symposium 14-19 June, pages 391–420. Academic Press, NY.

Wold, H. (1975). Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach. Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett.

Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems, 58(2):109–130.

Wu, T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. The Annals of Applied Statistics, 2(1):224–244.

Yohai, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. The Annals of Statistics, pages 642–656.

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

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. Journal of Computational and Graphical Statistics, 15:265–286.

Irene Hoffmann

Curriculum Vitae
October 2017

Contact Address
TU Wien
Institute of Statistics and Mathematical Methods in Economics
Research Group Computational Statistics

Wiedner Hauptstraße 8-10
A-1040 Vienna, Austria

Email: irene.hoffmann@tuwien.ac.at
Phone: +43 1 58801 105684

Career Summary

Since 10/2017	**TU Wien, Vienna (Austria)** Project assistant for *Upscaling deep buried geochemical exploration techniques into European business*
10/2014 - 09/2017	**TU Wien, Vienna (Austria)** Project assistant (FWF) for *Comet and Meteorite Materials - Studied by Chemometrics of Spectroscopic Data*
09/2016 - 01/2017	**BFI Wien, Vienna (Austria)** Lecturer for Quantitative Business Economics
03/2014 - 09/2014	**TU Wien, Vienna (Austria)** Research assistant in collaboration with BASF, New York
10/2013 - 01/2014	**BOKU, Vienna (Austria)** Lecturer at the Department for Applied Statistics and Computing
02/2013 - 12/2013	**ILF Consulting Engineers, Linz (Austria)** Internship: security and risk management Data modeling and statistical analysis of the risk of car accidents
07/2011 - 01/2012	**BOKU, Vienna (Austria)** Internship: Department for Bioinformatics Statistical evaluation of gene expression data

Education

10/2014-11/2017	**TU Wien, Vienna (Austria)** Doctor of technical sciences Thesis: *Sparse and robust modeling for high-dimensional data* PhD supervisor: Prof. Peter Filzmoser
02/2012 - 03/2014	**TU Wien, Vienna (Austria)** Master in Statistics Thesis: *Linear discriminant analysis for high-dimensional data*
04/2012 07/2012	**TU Berlin, Berlin (Germany)** ERASMUS exchange
08/2008 - 02/2012	**TU Wien, Vienna (Austria)** Bachelor in Statistics and Mathematics in Economics Thesis: *Sparse partial least squares for discrimination*

Publications

Hoffmann, I., Filzmoser, P., & Croux, C. Robust and sparse multigroup classification by the optimal scoring approach. Submitted for publication.

Ortner, T., Hoffmann, I., Filzmoser, P., Zaharieva, M., Breiteneder, C., & Brodinova, S. Multigroup discrimination based on weighted local projections. Submitted for publication.

Kurnaz, F. S., Hoffmann, I., & Filzmoser, P. Robust and sparse estimation methods for high-dimensional linear and logistic regression. Submitted for publication.

Hoffmann I., Filzmoser P., Serneels S., & Varmuza K. (2016). Sparse and robust PLS for binary classification. *Journal of Chemometrics*. 30(4):153–162.

Hoffmann I., Serneels S., Filzmoser P., & Croux C. (2015). Sparse partial robust M regression. *Chemometrics and Intelligent Laboratory Systems*, 149:50-59.

Serneels S. & Hoffmann I. (2015). sprm: Sparse and Non-Sparse Partial Robust M Regression and Classification. R package version 1.2.1. https://CRAN.R-project.org/package=sprm

Conference Talks

MOVISS - Metabolomic Bio & Data 2017, Vorau (Austria)
Title: Robust and sparse methods for high-dimensional linear and logistic regression.

Olomouc Days of Applied Mathematics (2017), Olomouc (Czech Republic)
Title: Inference for sparse and robust partial least squares regression.

9th International Conference of the ERCIM WG (2016), Seville (Spain)
Title: Robust and sparse classification by the optimal scoring approach.

Computer data analysis and modelling 2016, Minsk (Belarus)
Title: Sparse and robust classification by the optimal scoring approach.

International Conference on Robust Statistics 2016, Geneva (Swiss)
Title: Sparse and robust classification by the optimal scoring approach.

Conferentia Chemometrica 2015, Budapest (Hungary)
Title: Sparse and robust PLS for binary classification.

International Conference on Robust Statistics 2015, Kolkatta (India)
Title: Sparse and robust PLS for binary classification.

Posters

Varmuza K., Filzmoser P., Hoffmann I., Walach J., Cottin H., Fray N., Briois C., Silén J., Stenzel O., Kissel J., & Hilchenbach M (2017). Significance of variables for discrimination - applied to the search of organic ions in mass spectra measured on cometary particles. Poster at *Conferentia Chemometrica 2017*, Gyöngyös-Farkasmály (Hungary).

Varmuza K., Baklouti D., Bardyn A., Cottin H., Engrand C., Filzmoser P., Fray N., Hilchenbach M., Hoffmann I., Kissel J., Modica P., Silén J., Siljeström S., & Stenzel O. (2017) Comet dust composition explored by chemometric methods using mass spectral data from COSIMA/ROSETTA. Poster at the *15th Scandinavian Symposium on Chemometrics*, Naantali (Finland).

Varmuza K., Brandstätter F., Cottin H., Engrand C., Ferrière L., Filzmoser P., Fray N., Hilchenbach M., Hoffmann I., Kissel J., Koeberl C., Modica P., Paquette J., & Stenzel O. (2017) Elemental surface composition of comet 67P grains (Rosetta) and of carbonaceous chondrite meteorites - characterized by multivariate mass spectral data (COSIMA). Poster at the *European Geosciences Union General Assembly*, Vienna (Austria).

Varmuza K., Brandstätter F., Cottin H., Engrand C., Ferrière L., Filzmoser P., Fray N., Hilchenbach M., Hoffmann I., Kissel J., Koeberl C., Modica P., Paquette J., & Stenzel O. (2017). Comet and meteorite particle surface characterization by multivariate data analyses using TOF-SIMS data from COSIMA/Rosetta. Poster at *ANAKON 2017*, Gesellschaft Deutscher Chemiker, Tübingen (Germany).

Hoffmann I., Brandstätter F., Engrand C., Ferriére L., Filzmoser P., Hilchenbach M., Koeberl C. & Varmuza K. (2016). Meteorite classification by TOR-SIMS-chemometrics. Poster at *27th Mass Spec Forum*, Vienna (Austria).

Varmuza K., Hoffmann I., Filzmoser P., Brandstätter F., Engrand C., Ferrière L., Hilchenbach M., Koeberl C., Paquette J., Silén J. & Stenzel O. (2015). Recognition of relevant spectra in TOF-SIMS measurements on meteorite and comet grain samples by a chemometric approach. Poster at the *Conference on Solid State Analytics*, Vienna (Austria).

Varmuza K., Filzmoser P., Hilchenbach M., Hoffmann I., Kissel J. & Silén J. (2015). Selected chemometric approaches for mass spectra from comet dust grains (Rosetta). Poster at the *14th Scandinavian Symposium on Chemometrics*, Chia (Italy).