Robust Estimation and Variable Selection in High-Dimensional Linear Regression Models

by

David Kepplinger

B.Sc., Vienna University of Technology, 2012
Dipl.-Ing., Vienna University of Technology, 2015

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

August 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Robust Estimation and Variable Selection in High-Dimensional Linear Regression Models

submitted by **David Kepplinger** in partial fulfillment of the requirements for the degree of **Doctor of Philosphy** in **Statistics**.

Examining Committee:

Gabriela V. Cohen Freue, Statistics

Supervisor

Matías Salibián-Barrera, Statistics

Supervisory Committee Member

Alexandre Bouchard-Côté, Statistics

University Examiner

Anne Condon, Bioinformatics

University Examiner

Additional Supervisory Committee Members:

Ruben H. Zamar, Statistics

Supervisory Committee Member

Abstract

Linear regression models are commonly used statistical models for predicting a response from a set of predictors. Technological advances allow for simultaneous collection of many predictors, but often only a small number of these is relevant for prediction. Identifying this set of predictors in high-dimensional linear regression models with emphasis on accurate prediction is thus a common goal of quantitative data analyses. While a large number of predictors promises to capture as much information as possible, it bears a risk of containing contaminated values. If not handled properly, contamination can affect statistical analyses and lead to spurious scientific discoveries, jeopardizing the generalizability of findings.

In this dissertation I propose robust regularized estimators for sparse linear regression with reliable prediction and variable selection performance under the presence of contamination in the response and one or more predictors. I present theoretical and extensive empirical results underscoring that the penalized elastic net S-estimator is robust towards aberrant contamination and leads to better predictions for heavy tailed error distributions than competing estimators. Especially in these more challenging scenarios, competing robust methods reliant on an auxiliary estimate of the residual scale, are more affected by contamination due to the high finite-sample bias introduced by regularization.

For improved variable selection I propose the adaptive penalized elastic net S-estimator. I show this estimator identifies the truly irrelevant predictors with high probability as sample size increases and estimates the parameters of the truly relevant predictors as accurately as if these relevant predictors were known in advance. For practical applications robustness of variable selection is essential. This is highlighted by a case study for identifying proteins to predict stenosis of heart vessels, a sign of complication after cardiac transplantation.

High robustness comes at the price of more taxing computations. I present optimized algorithms and heuristics for feasible computation of the estimates in a wide range of applications. With the software made publicly available, the proposed estimators are viable alternatives to non-robust methods, supporting discovery of generalizable scientific results.

Lay Summary

This dissertation presents new methods for identifying variables, such as protein levels extracted from blood samples, relevant for predicting an outcome of interest, for example severity of a disease. The methods are specifically designed for applications where many variables are available, and the observed data possibly contains some highly unusual values. Examples of such unusual values are aberrantly high levels of some proteins in a blood sample, or an unusually severe disease outcome. These values can lead to biased and misleading results.

The methods proposed in this dissertation are less affected by unusual values and hence increase reliability of results. Therefore, results from a small set of observations are more likely to be generalizable to the broader population. The software is made openly available and gives researchers a versatile tool to support reliable scientific discoveries.

Preface

This dissertation is the original work of David Kepplinger, prepared under the supervision of Prof. Gabriela V. Cohen Freue.

Parts of Chapters 3 and 6 are based on a paper coauthored with the supervisor and two collaborators [G. V. Cohen Freue, D. Kepplinger, M. Salibián-Barrera, and E. Smucler (2019). "Robust elastic net estimators for variable selection and identification of proteomic biomarkers". In: *Annals of Applied Statistics* 13.4, pp. 2065–2090]. The original idea for the presented estimator and the procedure for initial estimates were developed by the supervisor and jointly refined by the author and supervisor. Development of algorithms, numerical experiments, and significant parts of manuscript writing were conducted by David Kepplinger. Other parts of the manuscript were jointly discussed by all coauthors, with significant contributions and developments by the author. The asymptotic properties presented herein are the original intellectual product of the author and pertain to conditions different than in the paper.

Table of Contents

List of Tables

List of Figures

Notation

Throughout this dissertation the following notation is consistently maintained. Chapter-specific notation is omitted here and defined where required.

Boldface characters denote vectors or matrices, whereas non-boldface characters are scalars. Capital characters in calligraphy typeface are reserved for random variables and random vectors, whereas observed values of random variables are written in regular typeface. Sets are denoted by capital characters in script typeface, e.g., $\mathcal{Q}$. The index variable i is only used to index observations in a sample, while j is reserved for indexing the set of predictors. Some commonly used symbols are

$\mathcal{Y}$	The random response variable in the linear regression model.
$\boldsymbol{\mathcal{X}}$	The random vector of predictors in the linear regression model.
$\mathcal{U}$	The random error term in the linear regression model.
y_i	The i-th observed response value.
$\mathbf{x}_i$	The vector of observed predictor values for the i-th observation.
x_{ij}	The value of the j-th predictor observed for the i-th observation.
$\mathbf{X}$	A matrix of observed predictor values, $\mathbf{X} = (\mathbf{x}_1^\intercal, \ldots, \mathbf{x}_n^\intercal)^\intercal$.
$\mathscr{Z}$	A sample, i.e., a set of observed values $\mathscr{Z} = \{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}$.

The boldface Greek letter $\boldsymbol{\beta}$ and the non-boldface Greek letter μ are reserved for the slope and intercept parameter, respectively, in the linear regression model. The boldface Greek letter $\boldsymbol{\theta}$ always denotes the concatenated vector of μ and $\boldsymbol{\beta}$, $\boldsymbol{\theta} = (\mu, \boldsymbol{\beta}^\intercal)^\intercal$. Accents, subscripts, and superscripts on $\boldsymbol{\theta}$ are propagated to μ and $\boldsymbol{\beta}$, e.g. $\hat{\boldsymbol{\theta}} = (\hat{\mu}, \hat{\boldsymbol{\beta}}^\intercal)^\intercal$. The total number of predictors in the linear regression model is denoted by p, i.e., the parameter vector

$\boldsymbol{\beta}$ has p elements, and the sample size is represented by n. Examples for such parameters are

$\boldsymbol{\beta}^0$	The true value of the slope parameter in the linear regression model.
μ^0	The true value of the intercept in the linear regression model.
$\hat{\boldsymbol{\theta}}$	Estimate of the intercept and slope parameters in the linear regression model.
β_j	The j-th element of a vector of slope coefficients.

Additionally, the following miscellaneous symbols and functions are often encountered in this dissertation:

$\mathbf{I}_n$	The identity matrix with n rows and n columns.
$\mathbf{1}_n$	A vector of n 1's.
$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^n$	The set of real vectors of dimension n.
$\mathbb{R}^{n\times p}$	The set of real matrices of dimension $n \times p$.
$\mathbb{E}_F$	The expected value with respect to distribution F.
$\|\cdot\|$	A vector or operator norm (if applied to a vector or a matrix, respectively).
$\boldsymbol{\nabla}_{\mathbf{u}} f(\mathbf{u})\Big\|_{\mathbf{u}=\tilde{\mathbf{u}}}$	The subgradient of function f with respect to $\mathbf{u}$, evaluated at $\tilde{\mathbf{u}}$.
$\mathscr{L}(\mathbf{y}, \hat{\mathbf{y}})$	A positive regression loss function taking values in $\mathbb{R}_+$, quantifying the difference between observed values $\mathbf{y} \in \mathbb{R}^n$ and fitted values $\hat{\mathbf{y}} \in \mathbb{R}^n$.
$\Phi(\boldsymbol{\beta})$	A penalty function $\mathbb{R}^p \to \mathbb{R}_+$, measuring the "size" of coefficients $\boldsymbol{\beta}$.
$\mathscr{O}(\boldsymbol{\theta})$	An objective function mapping regression coefficients in $\mathbb{R}^{p+1}$ to the set of positive real numbers.
$\hat{\boldsymbol{\theta}}_n \xrightarrow{a.s.} \boldsymbol{\theta}$	The random variable $\hat{\boldsymbol{\theta}}_n$ converges almost surely to $\boldsymbol{\theta}$ as the sample size n increases, i.e., $\Pr(\lim_{n\to\infty} \hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}) = 1$.
$\hat{\boldsymbol{\theta}}_n \xrightarrow{p} \boldsymbol{\theta}$	The random variable $\hat{\boldsymbol{\theta}}_n$ converges in probability to $\boldsymbol{\theta}$ as the sample size n increases, i.e., $\lim_{n\to\infty} \Pr(\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\| > \epsilon) = 0$ for any $\epsilon > 0$.

Glossary

The following acronyms are commonly used throughout. Each acronym is defined at its first occurrence in the text.

adaEN Adaptive elastic net

ADMM Alternating direction method of multipliers

CAV Cardiac allograft vasculopathy

CV Cross-validation

DAL Dual augmented Lagrangian

EN Elastic net

FBP Finite-sample breakdown point

LASSO Least absolute shrinkage and selection operator

LARS Least angle regression

LS Least squares

LAD Least absolute deviation

LOO Leave-one-out

MAD Median absolute deviation

MSE Mean-square error

PENSE Penalized elastic net S-estimator

PSC Principal sensitivity component

PVE Percentage of variance explained

PY Peña-Yohai procedure

RCV Refitted cross-validation

RMSPE Root mean-square prediction error

Acknowledgements

My journey through the PhD program would not have been as successful without the support of truly remarkable people. I am grateful for the guidance by inspiring mentors, first and foremost by my supervisor Dr. Gabriela Cohen Freue. Her dedication and support has helped me to get where I am today, professionally and personally. Giving me autonomy while ensuring I would not lose sight on what's important has allowed me to grow as a scholar and person. Her advice at professional and personal crossroads has been a godsend and working together for the past five years has been nothing but inspiring. Of course, the members of my supervisory committee have also been instrumental to this dissertation and my development at UBC. Thanks to Dr. Matías Salibián-Barrera for his insightful input and the many stimulating discussions. I also thank Dr. Ruben Zamar for sharing his profound expertise and providing me with the opportunity to gain research experience in industry.

I count myself very lucky to have been part of the Department of Statistics which has provided me with a rich learning experience and highly supportive environment. I greatly appreciate Peggy Ng, Ali Hauschildt and Andrea Sollberger for their lifting spirits even during stressful times and their grokking of the UBC apparatus. They were never too busy to ask about my well-being and lending an open ear. Department-organized seminars, department teas, and grad trips have provided opportunities to foster connections and friendships. I am grateful to many other members of the department who I have been so privileged to meet and who have shared their vast experience on numerous occasions, such as Melissa Lee, Dr. Nancy Heckman, Dr. John Petkau, Dr. Paul Gustafson, and Dr. Will Welch, among others. Furthermore, volunteer positions made available by the department, such as graduate student representative and membership on search committees, have given me valuable leadership skills and insights into academic processes. My studies have only been made possible by generous funding from UBC through the Four-Year Fellowship and Faculty of Science Graduate Awards. Dr. Cohen Freue and Dr. Zamar have also graciously supported my studies through research assistantships. I could not have wished for a better

environment to pursue my doctorate.

I would have never made it to this point without the incredible support from my family. My gratitude for my spouse Alexandra Patzak is immeasurable, for she has been a constant source of inspiration and energy. I cannot find the words to express how lucky I am to have her in my life and to have embarked on our PhD journeys together. I will also forever be thankful to my mother Heidi and sister Sara for inciting and nurturing my curiosity, teaching me the importance of education and candor, and for staying close to me (at least virtually) wherever life took me. So did my grandmother Elisabeth, in her joyful way, although I know she would have preferred me to stay closer to home. Her cryptic recipes with little instructions have kept me well fed over the years. My uncle Dietmar I thank for his take on teaching and the side-projects which have been a welcoming contrast to research. I thank all of them for their efforts to follow my research, despite my struggles explaining it in an accessible manner.

I will always cherish the memories I have made with the many uniquely wonderful people I have come to know during my time at UBC. I will fondly remember the subtle sense of humor of Eric Fu when he was updating me on everything that was going on in the department. I have also very much enjoyed my conversations with Creagh Briercliffe, who has never felt obliged to hide his sarcastic, refreshingly indecorous, wit. Daniel Hadley's incisive analyses of society were thought-provoking, yet he managed that we (almost) always ended up laughing. One of my proudest moments at UBC, thanks to our captain Daniel Dinsdale, was winning the UBC departmental futsal division alongside fabulous teammates Jonathan Agyeman, Jonathan Steif, Joe Watson, and Dr. Matías Salibián-Barrera. Although Derek Cho left for a job in Japan before our victorious campaign, I also credit him for the win, and thank him for explaining peculiarities of Vancouver's culture or our conversations about hockey. I thank Andy Leung for his inspiring integrity, for standing up for his friends and colleagues, and for introducing me to exciting new flavors of Japanese and Chinese cuisine. I have met numerous more people who have made my PhD studies unforgettable, such as Sonja Isberg and Vincenzo Coia, among many others.

All of these people whose paths overlapped with mine over the past five years and more made a unique impression on my dissertation and shaped me as a person. For this I am forever grateful. Thank you all.

To Alex, whose laughter
is my medicine.

Chapter 1

Introduction

The ability to predict a continuous response of interest using a set of predictors is central to many scientific and industrial applications. Technological advances significantly pushed the frontiers in science and industry by enabling the collection of immense numbers of possibly relevant predictors. The scientific goal is two-fold: (a) predicting the response if only the predictors' values are available and (b) identifying which predictors are relevant, particularly for accurate prediction. The relationship at the heart of the problem may be highly complex, but a crude approximation by a linear relationship using a small set of the available predictors can nevertheless give valuable insights into the involved processes and allow for accurate prediction of the response. Approximations by a linear model are pertinent in applications where the sample size is small, particularly if many predictors are available. As an example, consider predicting yield of a crop based on numerous predictors such as a variety's genotype, nutrition content of the soil, and other environmental factors. Collecting a large sample is complicated by several obstacles, such as the time required to fully grow the crop, costs of measuring all possible predictors, but also continued cooperation of growers who are willing to share their trade secrets.

The scientific goal translates to a statistical goal of estimating the parameters relating the values of the predictors with the response, with emphasis on identifying which coefficients are truly non-zero. Assuming that only some of the available predictors are relevant leads to the linear regression model being sparse in the sense that only these few relevant predictors have non-zero coefficient. A myriad of methods is available for estimating parameters and simultaneously identifying relevant predictors in the sparse linear regression model. These methods are predominantly founded in the assumption that all observations in the sample at hand are equally trustworthy. The danger of this assumption is that even

a single contaminated observation, for instance an observation with aberrant response value and/or highly anomalous values in one or more predictors, can jeopardize the reliability of these methods and, in turn, the generalizability of the estimated predictive model. Contamination can take countless forms, but contaminated observations are generated by unknown processes different from the linear model underlying the majority of observations. The more predictors are available, the more questionable the assumption of no contamination in the sample at hand is, thereby exacerbating the risk of spurious discoveries.

Robust methods for linear regression, in contrast, are devised to cope with potential presence of contamination. Robust methodology for problems with only a small number of predictors is well established, but these solutions are challenged by characteristics inherent to high dimensional data and the notion of sparse models. Therefore, robust methods for simultaneous estimation and variable selection have not seen the same proliferation as non-robust methods. One of the biggest roadblocks to applying robust methods in high dimensional problems is computational complexity. Computation of robust estimates is difficult even in low dimensional settings, but as dimensions grow computational complexity can become insurmountable. Furthermore, robust methods are devised under the assumption that some of the observations may be contaminated. This inherent "mistrust" leads in general to less precise parameter estimates compared to non-robust methods in pristine settings without contamination. To compensate for this loss of efficiency, robust methods are often two-tiered: first computing a highly robust but potentially imprecise estimate and then refining this estimate to gain precision. The refinement step, however, is problematic in high dimensions and can, in worst case scenarios, lead to the loss of robustness and hence reliability. Last but not least, the interplay of sparsity and possible contamination adds a layer of difficulty which received little attention in the existing literature.

The first two contributions of this dissertation are the development and study of two robust estimators for high dimensional sparse linear regression. Both estimators are highly robust towards possibly large amounts of contamination in the data and perform reliably even in the most challenging situations where two-tiered robust estimators are at an elevated risk of being unduly affected by contamination. Understanding the interaction between sparsity and possible contamination provides important insights into its effect on estimators' ability to identify these relevant predictors. Particularly in very sparse problems, i.e., where only a small number of the available predictors are truly relevant for prediction, one of the proposed robust estimators protects against an inflation of the number of irrelevant predictors wrongly selected due to contamination.

This work also sheds light onto the difficulty of performing refinement steps to improve precision of robust estimators in high dimensional sparse problems. While justified theoretically for estimators without sparsity constraints, the theoretical foundation of the refinement step crumbles when sparsity is induced. Furthermore, robustness of the refinement step is contingent on an accurate estimate of the residual scale but obtaining this estimate in high dimensions under contamination is difficult. Therefore, applying the refinement step in high dimensions may jeopardize the reliability of the estimator.

The final main contribution is the adaptation and implementation of algorithms for computing the proposed robust estimators of linear regression. Robust estimation poses several computational challenges inherent to taming the influence of potentially contaminated observations. These challenges are especially taxing in the high dimensional problems considered in this work. Analysis and rigorous optimization of the developed algorithms curtails computational complexity and ensures feasibility of robust estimation in a wide range of applications. These algorithms are made publicly available through an accessible software package, paving the way for robust estimation to take a foothold in high dimensional data analysis.

Broadly speaking this dissertation is concerned with robust estimation in high dimensional linear regression models under the assumption that only some of the numerous available predictors are truly relevant for prediction, also known as the sparsity assumption. The specific focus is on simultaneous parameter estimation and variable selection through penalizing the size of non-zero coefficients, known as regularized estimation. Chapter 2 gives a comprehensive summary of the sparse linear regression model, the effects of contamination, and robust estimation in low dimensional settings. The chapter continues by outlining the benefits of the sparsity assumption in high dimensional linear regression and how regularization induces sparsity in estimates and concludes with an overview of avenues for fusing the sparsity assumption and robust estimation.

In Chapters 3 and 4, I present two robust regularized estimators for sparse linear regression. The estimator presented in Chapter 3, the penalized elastic net S-estimator (PENSE), combines robust estimation via the S-loss for linear regression (Rousseeuw and Yohai 1984) with the elastic net penalty (Zou and Hastie 2005) for variable selection. The chapter delineates an elaborate scheme germane to locating global optima of the non-convex PENSE objective function and establishes theoretical properties pertaining to robustness and asymptotic consistency of the estimator, highlighting its reliability even under challenging circumstances. Theoretical results, however, lack guidance for selecting hyper-parameters intro-

duced with the elastic net penalty, governing sparsity and prediction performance of the ensuing estimate. I discuss strategies for selecting hyper-parameters in practical applications and ascertain favorable finite-sample performance of PENSE in an extensive simulation study. While empirical results underline the good prediction performance of PENSE, they also expose shortcomings in its ability to screen out irrelevant predictors.

To improve upon the high false positive rate of PENSE, Chapter 4 introduces the adaptive PENSE estimator, combining the robust S-loss with the adaptive elastic net penalty (Zou and Zhang 2009). Leveraging a preliminary PENSE estimate to penalize predictors differently, adaptive PENSE is shown to possess the oracle property even under adverse conditions. Asymptotically, the adaptive PENSE estimator correctly identifies all truly irrelevant predictors with high probability and estimates the non-zero coefficients for the truly relevant predictors as efficient as if they were known in advance. Importantly, variable selection by adaptive PENSE is highly resilient against aberrant values in the truly irrelevant predictors, whereas PENSE and other robust regularized estimators would falsely identify the affected predictors as relevant. The improved robustness of variable selection is important for practical applications. This is demonstrated by applying adaptive PENSE to biomarker discovery for cardiac allograft vasculopathy, a common complication in heart transplant recipients.

A common strategy for obtaining more accurate robust estimators is to refine a highly robust, but possibly imprecise estimate. The strategy is successful in low dimensions but proves less reliable in higher dimensions. The refinement step hinges on a robust scale of the residuals from the initial, highly robust, fit. As Chapter 5 outlines, robust estimation of the residual scale faces several challenges in high dimensions. While PENSE, adaptive PENSE, and other highly robust regularized estimators perform well for prediction, the empirical distribution of the residuals and robust estimates of the residual scale are severely biased in finite samples with many predictors. The inflated bias can hamstring the refinement step or, worse, make it susceptible to the influence of contamination. I present empirical results demonstrating that existing remedies developed for de-biasing non-robust residual scale estimates do not work well for robust estimates. This underlines the practical importance of robust regularized methods which do not depend on robust estimates of the residual scale, such as PENSE and adaptive PENSE.

The estimators proposed in this dissertation incur multiples of the computational costs of comparable non-robust estimators. The algorithms and heuristics detailed in Chapter 6 are therefore paramount for ensuring applicability of the estimators to high dimensional

problems. I adapt several algorithms for minimizing convex functions such that they can be utilized to efficiently locate minima of non-convex robust objective functions. With this variety of algorithms the range of problems amenable to robust regularized estimators is expanded, enabling the use of (adaptive) PENSE in a wide range of problems. Especially in conjunction with the need to select appropriate hyper-parameters, computational complexity would balloon without the optimized algorithms developed for PENSE and adaptive PENSE.

Chapter 2

Background

In this chapter I formally introduce the linear regression model and outline several methods to estimate the parameters in this model. I expose how some estimators of linear regression are affected even by minor contaminations and in Section 2.2 I outline common strategies to derive estimators that are robust against these contaminations. For applications where it can be assumed that many of the available predictors are truly unrelated with the response the methods in Section 2.2 are suboptimal. In Section 2.3 I discuss methods to estimate the regression parameters while also identifying those "irrelevant" predictors and shed some light on possible improvements in the presence of contamination.

2.1 The Linear Regression Model

As outlined in Chapter 1, the linear regression model discussed in this work assumes that the value of a response variable $\mathcal{Y}$ (taking values in $\mathbb{R}$) relates to the values of a random vector of predictors $\boldsymbol{\mathcal{X}}$ (taking values in $\mathbb{R}^p$) through a linear function of the form

$$\mathcal{Y} = \mu^0 + \boldsymbol{\mathcal{X}}^\intercal \boldsymbol{\beta}^0 + \mathcal{U} \tag{2.1}$$

where $\mu^0 \in \mathbb{R}$ and $\boldsymbol{\beta}^0 \in \mathbb{R}^p$ are the true, unknown regression parameters, and $\mathcal{U}$ is a random error following some distribution F_0. To make the arguments in this work more concise, $\boldsymbol{\theta}^0 \in \mathbb{R}^{p+1}$ denotes the concatenated parameter vector $\left(\mu^0, \boldsymbol{\beta}^{0\intercal}\right)^\intercal$.

I assume that the random predictor vector $\boldsymbol{\mathcal{X}}$ is independent of $\mathcal{U}$ and follows distribution H_0. Therefore, the joint distribution G_0 of $(\mathcal{Y}, \boldsymbol{\mathcal{X}})$ factorizes into the product

$$G_0(y, \mathbf{x}) = H_0(\mathbf{x}) F_0(y - \mu^0 - \mathbf{x}^\intercal \boldsymbol{\beta}^0). \tag{2.2}$$

It is important to highlight that so far, the only assumptions on the distributions is that $\mathcal{U}$ is centered at zero and that $\boldsymbol{\mathcal{X}}$ and $\mathcal{U}$ are independent.

Without any additional assumptions, the linear regression model (2.1) can be used to relate the conditional expectation of the response to the predictors through a linear function. Assuming the expected value $\mathbb{E}_{F_0}[\mathcal{U}] = 0$, independence of F_0 and H_0 leads to an expression of the conditional expectation of the response in the form of

$$\mathbb{E}_{F_0}[\mathcal{Y}|\boldsymbol{\mathcal{X}} = \mathbf{x}] = \mu^0 + \mathbf{x}^\intercal \boldsymbol{\beta}^0. \tag{2.3}$$

If the parameters are known, this expression can be used to predict the value of the response which can be expected given only observed values of the predictors.

In practice the true parameters are of course unknown. Using (2.3) for predicting the response based only on observed values of the predictors therefore requires estimates of the parameters. For estimating these parameters, it is assumed a sample of $n > 0$ independent realizations of $(\mathcal{Y}, \boldsymbol{\mathcal{X}})$ is available. The observed sample is written as the vector-matrix pair $(\mathbf{y}, \mathbf{X})$, where $\mathbf{y} = (y_1, \ldots, y_n)^\intercal$ and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal$. The observed response values $y_i \in \mathbb{R}$ and associated observed predictor values $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, are used to compute estimates of the parameters according to some estimation method. The quality of these estimates and thus the prediction can be assessed by analyzing the statistical properties of the estimator, i.e., the random vector arising from applying the estimation method to the random sample $(\mathcal{Y}_i, \boldsymbol{\mathcal{X}}_i)$, $i = 1, \ldots, n$.

An important quality of an estimator is for the estimate to "be close" to the true parameter value. Ideally, an estimator should be unbiased, $\mathbb{E}_{G_0}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}^0$, and have small variance, $\mathbb{E}_{G_0}[\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\|_2^2]$. Tolerating a small bias in finite-samples, however, can often lead to an estimator with smaller variance. More important than unbiasedness is that both bias and variance vanish as the sample size increases. This is the case if the estimator is consistent for the true parameter, $\lim_{n\to\infty} \mathbb{P}(\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\| > \epsilon) = 0$ for every $\epsilon > 0$, or even strongly consistent, $\mathbb{P}(\lim_{n\to\infty} \hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^0) = 1$. A consistent estimator may be biased in finite samples but its bias and variance tend to 0 as the sample size increases.

Having a consistent estimator $\hat{\boldsymbol{\theta}}$ and being able to derive the asymptotic distribution of $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0)$ enables statistical inference on the parameters and comparisons between estimators. Of particular interest are estimators converging to a Normal distribution with mean 0 and covariance matrix $\mathbf{V}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$ which can be factorized into $\upsilon(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)\tilde{\mathbf{V}}(\boldsymbol{\theta}^0)$, where $\upsilon(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) \in \mathbb{R}$. In case two estimators $\hat{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\theta}}$ converge to such a Normal distribution, they

can be compared by the ratio of $v(\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$ to $v(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$, i.e., the asymptotic relative efficiency of $\hat{\boldsymbol{\theta}}$. Usually, $\tilde{\boldsymbol{\theta}}$ is taken to be an estimator with small variance in a particular setting, e.g., the maximum likelihood estimator (MLE), if it exists. Asymptotic relative efficiency is useful for quantifying the costs incurred by an estimator $\hat{\boldsymbol{\theta}}$ which, for example, requires less stringent assumptions on the model than $\tilde{\boldsymbol{\theta}}$.

Asymptotic properties facilitate comparison between estimators but give limited insights into the estimator's qualities when the sample size n is small. Finite-sample properties, on the other hand, are more useful assessments of the performance of an estimator in practical applications, but at the same time are difficult to derive theoretically, except for in a few special cases. For many regression estimators and model distributions G_0, finite-sample performance measures are therefore calculated through extensive simulations. With prediction performance being of primary interest in this work, the mean squared prediction error (MSPE) is an important measure of performance in finite-samples:

$$\begin{aligned} \mathrm{MSPE}(\hat{\boldsymbol{\theta}}, G_0) &:= \mathbb{E}_{G_0}\left[\left(\tilde{\mathcal{Y}} - \left(\hat{\mu} + \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}\right)\right)^2\right] \\ &= \mathrm{Var}_{G_0}\left[\tilde{\mathcal{Y}} - \hat{\mu} - \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}\right] + \mathbb{E}_{G_0}\left[\tilde{\mathcal{Y}} - \hat{\mu} - \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}\right]^2. \end{aligned} \tag{2.4}$$

Here, the expectation is taken over the n observations in the sample used to estimate $\hat{\boldsymbol{\theta}}$ as well as the "new" observation $(\tilde{\mathcal{Y}}, \tilde{\boldsymbol{\mathcal{X}}})$. The mean squared prediction has the intuitive interpretation of the sum of the variance of the prediction error $\tilde{\mathcal{Y}} - \hat{\mu} - \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}$ and its squared bias. It can therefore be seen as an overall metric of prediction performance.

The MSPE can also be written as

$$\begin{aligned} \mathrm{MSPE}(\hat{\boldsymbol{\theta}}, G_0) &= \mathbb{E}_{G_0}\left[\left(\tilde{\mathcal{Y}} - \left(\hat{\mu} + \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}\right)\right)^2\right] \\ &= \mathbb{E}_{G_0}\left[\left(\mu^0 + \tilde{\boldsymbol{\mathcal{X}}}^\intercal \boldsymbol{\beta}^0 + \tilde{\mathcal{U}} - \hat{\mu} + \tilde{\boldsymbol{\mathcal{X}}}^\intercal \hat{\boldsymbol{\beta}}\right)^2\right] \\ &= \mathbb{E}_{G_0}\left[\tilde{\mathcal{U}}^2\right] + 2\mathbb{E}_{G_0}\left[\tilde{\mathcal{U}}\right] \mathbb{E}_{G_0}\left[(\mu^0 - \hat{\mu}) + \tilde{\boldsymbol{\mathcal{X}}}^\intercal (\boldsymbol{\beta}^0 - \hat{\boldsymbol{\beta}})\right] \\ &\quad + \mathbb{E}_{G_0}\left[\left((\mu^0 - \hat{\mu}) + \tilde{\boldsymbol{\mathcal{X}}}^\intercal (\boldsymbol{\beta}^0 - \hat{\boldsymbol{\beta}})\right)^2\right]. \end{aligned}$$

The first term in the last line is the variance of the errors and the middle term is 0 because the errors $\tilde{\mathcal{U}}$ are centered and independent of the predictors. The final term is the mean

squared error (MSE) of the estimator, defined by

$$\begin{aligned}\mathrm{MSE}(\hat{\boldsymbol{\theta}}, G_0) := \mathbb{E}_{G_0}\left[(\hat{\mu} - \mu^0)^2\right] + 2\mathbb{E}_{G_0}\left[(\hat{\mu} - \mu^0)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal\right]\mathbb{E}_{H_0}[\tilde{\boldsymbol{\mathcal{X}}}] \\ + \mathbb{E}_{G_0}\left[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal \mathbb{E}_{H_0}[\tilde{\boldsymbol{\mathcal{X}}}\tilde{\boldsymbol{\mathcal{X}}}^\intercal](\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)\right].\end{aligned} \tag{2.5}$$

This definition of the MSE from a prediction-based perspective (Maronna et al. 2019) measures the overall estimation accuracy, taking into account the covariance among predictors and their multivariate location. Comparable to the asymptotic relative efficiency, the finite-sample efficiency of an estimator $\hat{\boldsymbol{\theta}}$, defined as $\mathrm{MSE}(\tilde{\boldsymbol{\theta}}, G_0)/\,\mathrm{MSE}(\hat{\boldsymbol{\theta}}, G_0)$, facilitates comparison of estimation accuracy between different estimators. Again, the estimator $\tilde{\boldsymbol{\theta}}$ is a "gold standard", e.g., the maximum likelihood estimator as defined below, and finite-sample efficiency is desirable to be close to or even larger than 1.

Closely related to the MSE, the L_2 estimation error $\mathbb{E}_{G_0}\left[\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\|_2^2\right]$ provides similar information about the finite-sample performance of an estimator. The L_2 estimation error, however, ignores the covariance among predictors, i.e., omitting $\mathbb{E}_{H_0}[\tilde{\boldsymbol{\mathcal{X}}}\tilde{\boldsymbol{\mathcal{X}}}^\intercal]$ in (2.5). The MSE and the L_2 estimation error coincide if the predictors are centered and pairwise independent with identical variance. In cases where predictors are highly correlated, the MSE remains small even if the parameter estimates are slightly biased, as long as the combined effect of the correlated predictors (i.e., the sum of the scaled coefficient values) is close to the truth. As an example, consider a linear regression model with two centered predictors which are highly correlated (e.g., $\mathrm{Cor}_{H_0}(\mathcal{X}_1, \mathcal{X}_2) \approx 1$) and have variance σ_1^2 and σ_2^2, respectively. In this case, the MSE is small as long as $\hat{\beta}_1\sigma_1 + \hat{\beta}_2\sigma_2 \approx \beta_1^0\sigma_1 + \beta_2^0\sigma_2$. Considering that both $\mathcal{X}_1$ and $\mathcal{X}_2$ carry almost the same information, the actual value of the parameters is irrelevant for explaining the response well, as long as the sum of the scaled coefficients is close to the truth. For the L_2 estimation error to be small, on the other hand, both $|\hat{\beta}_1 - \beta_1^0|$ and $|\hat{\beta}_2 - \beta_2^0|$ must be small.

Even when restricting attention to estimators that possess several of the above listed desired properties, there is a plethora of methods available to estimate the regression parameters in the linear regression model. Which method to use depends on the researcher's emphasis as well as additional assumptions that can be imposed. For instance, if the distribution of the errors is (assumed to be) known to have density function f_0, the maximum likelihood estimator (MLE)

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \underset{\mu,\boldsymbol{\beta}}{\arg\min} \sum_{i=1}^{n} -\log f_0\left(y_i - \mu - \mathbf{x}_i^\intercal\boldsymbol{\beta}\right)$$

has very appealing properties as the sample size n grows. Under mild regularity conditions on the distribution F_0 and as $n \to \infty$, the MLE is consistent (i.e., converges to the true parameters in probability) and asymptotically efficient (i.e., no consistent estimator can have lower variance). However, these optimality properties heavily depend on the validity of the assumption on G_0.

A different approach to estimate the parameters is by trying to fit the observed response values $\mathbf{y}$ well without regard of the actual distribution of the errors. Formally, the approach is to determine $\hat{\boldsymbol{\theta}}$ such that

$$\hat{\boldsymbol{\theta}} = \arg\min_{\mu,\boldsymbol{\beta}} \mathscr{L}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}), \tag{2.6}$$

where the regression loss function $\mathscr{L}\colon \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$ measures the inaccuracy of the fitted values, i.e., how far the fitted values $\hat{\mathbf{y}} = \hat{\mu} + \mathbf{X}\hat{\boldsymbol{\beta}}$ are from the observed response $\mathbf{y}$. Therefore, it makes sense to require that $\mathscr{L}(\mathbf{y}, \hat{\mathbf{y}}) = 0$ if and only if $\hat{\mathbf{y}} = \mathbf{y}$. Paraphrasing Lehmann and Casella (2003), the desire is to have an accurate estimate, but since it is usually unknown what the estimate will be used for once it is made public, the choice of the measure of accuracy is arbitrary. However, the chosen loss function directly affects the properties of the estimator and therefore it should be chosen wisely. The most prominent loss function for linear regression is the sum of square residuals

$$\mathscr{L}_{\text{LS}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

which is mathematically convenient and leads to an accurate and theoretically sensible estimator in many settings. In the case where F_0 is assumed Gaussian, for instance, the least squares (LS) estimator, $\hat{\boldsymbol{\theta}}_{\text{LS}}$ coincides with the MLE and thus enjoys all the asymptotic properties of the MLE. Even more convincing, the Gauss-Markov theorem (and extensions of it) states that the LS-estimator has uniformly smallest variance among all unbiased linear estimators if (i) the variance of the error term is finite and (ii) the distribution of the predictors H_0 is unknown, or G_0 is multivariate Normal with unknown parameters (Lehmann and Casella 2003, p. 184f).

Despite these strong arguments for the LS-estimator, there are reasons why the LS-estimator might not be the best choice. The LS-estimator has smallest variance among all unbiased and linear estimators (i.e., estimators for which the fitted values are a linear combination of the observed response values). Consequently, unless F_0 is Gaussian, it may

be possible to find an estimator that is not linear in the observed response values or biased (but still consistent) and has smaller variance than the LS-estimator.

Especially if it is likely that the error term takes on large values, i.e., F_0 has heavy tails, finite-sample performance of the LS-estimator suffers considerably. Even if the researcher is willing to assume the error term is Normally distributed, it is most often only a crude approximation to the truth and large errors may occur more often than expected. Because of the square function, unusually large residuals contribute substantially to the LS-loss and force the estimator $\hat{\boldsymbol{\theta}}_{\text{LS}}$ to adapt to these observations to shrink the discrepancy between the fitted value and the observed value. If the sample at hand contains a small fraction of observations with unusually large residuals, they can dominate the LS-loss function and the estimate could be excessively affected by them.

A maybe even more worrisome property of the LS-loss is revealed when considering its gradient with respect to the regression parameters, which is given by

$$\nabla_{\boldsymbol{\beta}} \mathscr{L}_{\text{LS}}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta})\Big|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = -\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu - \mathbf{x}_i^\intercal \tilde{\boldsymbol{\beta}})\mathbf{x}_i.$$

At every minimum of the LS-loss, each element of the gradient needs to be 0. Therefore, the LS-estimator $\hat{\boldsymbol{\theta}}_{\text{LS}}$ must satisfy

$$\mathbf{0}_p = \sum_{i=1}^{n}(y_i - \hat{\mu}_{\text{LS}} - \mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}_{\text{LS}})\mathbf{x}_i,$$

where $\mathbf{0}_p$ is the p-dimensional 0-vector. From this equation it can be clearly seen that if the value of any predictor of the i-th observation is unusually large, the corresponding response needs to be fitted very well to keep the residual small and counterbalance the influence of the predictor on the gradient. Observations with unusually large values in any of the predictors are called leverage points; unless the true residual of this observation is very small, the observation can have a devastating effect on the LS-estimator. Huber and Ronchetti (2009) argue leverage points are usually of no concern in designed experiments where the researcher has (at least some) control over the values of the predictors. Even with random predictors, as considered in this work, Huber and Ronchetti suggest that leverage points are interesting by themselves and should be identified in advance to be analyzed separately. This approach might work in some settings, but in Section 2.3 I argue why this is challenging or nearly impossible in the settings considered in this work.

Now that I have outlined some instances where the LS-loss might not be an appropriate

choice, the question becomes if there are alternatives with similar appealing properties. Of course, one possibility is to assume a different distribution for the errors, one with heavier tails, and compute the MLE. However, this approach might lose precision if a large majority of the observations are well explained by a regression model where F_0 has light tails (e.g., Gaussian) and only a few observations have gross errors. Additionally, the MLE does not address the problem of leverage points. In the next section I introduce a strategy from robust statistics.

2.2 Robust Estimation in the Linear Regression Model

In many practical applications of linear regression, precluding the presence of adverse observations is almost impossible. The approach taken by robust statistics is to not try to build a comprehensive model that accounts for these few adverse observations, but rather derive methods that are stable and give "reasonable" results as long as the number of adverse observations remains small. Importantly, it is assumed that the parametric model G_0 underlies the majority of the observations in the sample. However, to allow for a more realistic representation of the observed sample, a small proportion of the sample is allowed to come from an unspecified, possibly degenerate, model $\breve{G}$. In the Tukey-Huber contamination model for linear regression, this can be written as $\tilde{G}_0(y, \mathbf{x}) = (1 - \epsilon)G_0(y, \mathbf{x}) + \epsilon\breve{G}(y, \mathbf{x})$, with G_0 the parametric model defined in (2.2) and contamination proportion $\epsilon \in [0, 0.5)$. In this "casewise" contamination model, the observed sample is generated by a mixture of the data generating process of interest, G_0, and the contamination process $\breve{G}$. The goal is still to estimate the parameters in G_0, but it is more difficult because some of the observations are actually generated by $\breve{G}$, and it is not known which observations. An observation is only useful for estimating the parameters if it is indeed generated by G_0 and robust procedures designed for the Tukey-Huber should filter information from observations generated by $\breve{G}$. To ensure G_0 and hence the parameters in the model are identifiable, the contamination proportion ϵ should be less than 50%, i.e., the majority of the observed sample is generated from the process of interest.

The casewise contamination model can be compared to the more general independent contamination model (Alqallaf et al. 2009) where each individual value of the observation is independently either generated by the assumed model or by the unspecified contamination process. If thinking of the sample as an $n \times (p + 1)$ matrix, with the i-th observation being recorded in the i-th row and the j-th column corresponding the value of the j-th

predictor (or the response if $j = p + 1$), the independent contamination model can be thought of as "cellwise" contamination. In this framework, each cell is either generated by the true model, or by the unspecified contamination process. In the casewise contamination model, on the other hand, each observation with a single contaminated value is considered to be generated by $\breve{G}$. Having a few contaminated cells can lead to a large number of contaminated cases, especially if p is large. This may be problematic in high-dimensional datasets as the proportion of contaminated cases could be propelled outside the sustainable 50%. The cellwise contamination model, however, poses great challenges for estimation procedures which go beyond the scope of this work. Henceforth contamination is always understood in the sense of the Tukey-Huber contamination model, i.e., an observation is either considered contaminated or not.

Despite the presence of a small proportion of contamination, the aim is still to estimate the true regression parameters in G_0, $\boldsymbol{\theta}^0 = (\mu^0, \boldsymbol{\beta}^0)$. However, in addition to the desired properties for any estimator discussed in the previous section, robust estimators strive to limit the effect of adverse observations. Over time, different measures of robustness, and thereby properties related to these measures, have been developed. A concept that plays a central role in this work is the notion of the replacement finite-sample breakdown point (FBP) as defined in Donoho and Huber (1982). The FBP measures how many observations in any given sample must be replaced by arbitrary values to push the estimate to the boundary of the parameter space. In the context of regression, this is equivalent to forcing the norm of the estimated regression parameter to infinity. To define the breakdown point formally, I introduce the notation $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\mathscr{Z})$ for an estimator of the regression parameters to explicitly show the dependence on the sample $\mathscr{Z} = (\mathbf{y}, \mathbf{X}) = \{(y_i, \mathbf{x}_i) \colon i = 1, \ldots, n\}$. With this notation, an estimate of the regression parameters has FBP $\epsilon^*(\hat{\boldsymbol{\theta}}, \mathscr{Z})$ given by

$$\epsilon^*(\hat{\boldsymbol{\theta}}, \mathscr{Z}) = \max_{m=1,\ldots,n} \left\{ \frac{m}{n} \colon \sup_{\tilde{\mathscr{Z}} \in \mathfrak{Z}_m(\mathscr{Z})} \|\hat{\boldsymbol{\theta}}(\tilde{\mathscr{Z}})\| < \infty \right\}. \tag{2.7}$$

The set $\mathfrak{Z}_m(\mathscr{Z})$ denotes all possible samples obtained by replacing at most m observations from the original sample $\mathscr{Z}$ with arbitrary values. Ideally, the FBP does not depend on the actual sample $\mathscr{Z}$, as long as the sample satisfies some estimator-dependent conditions. The FBP can be considered as a measure of how much contamination can be tolerated without suffering the worst-possible consequences. It does not, however, imply that for less contamination the estimate is anywhere close to the true parameter $\boldsymbol{\theta}^0$; this includes that the estimator does not have to be consistent for $\boldsymbol{\theta}^0$ under any positive amount of

contamination. A related concept is the (asymptotic) breakdown point which, instead of operating on the sample level, considers the worst effect on the parameter estimate if the actual distribution $\tilde{G}_0$ is within an ϵ neighborhood of the assumed distribution G_0 (Davies and Gather 2005). A driving factor in the development of many robust estimators is the desire to obtain a "high breakdown point" estimator; i.e., an estimator that achieves a breakdown point close to 0.5, the maximum for regression-equivariant estimators [1] (Davies and Gather 2005).

Instead of focusing on the worst-case scenarios, the sensitivity curve measures how much the parameter estimate changes when adding a single observation to the original sample (Maronna et al. 2019). The asymptotic version of the sensitivity curve, the influence function, measures the effect on the estimate when adding infinitesimal point-mass at $(\tilde{y}, \tilde{\mathbf{x}})$ to the assumed distribution G_0 (Hampel 1974). In general, it is desired that a robust estimator has a bounded sensitivity curve and influence function. However, even if an estimator has a breakdown point greater than 0, neither the sensitivity curve nor the influence function needs to be bounded.

A more balanced measure is the maximum asymptotic bias (MB) which measures by how far a consistent estimator misses the target value $\boldsymbol{\theta}^0$ if the actual distribution $\tilde{G}_0$ is in an ϵ-neighborhood of the assumed G_0 (Maronna et al. 2019). The maximum bias gives a more refined picture of how badly an estimator can be affected by a certain amount of contamination and from the definition of the breakdown point it is evident that the MB is finite for $\epsilon \leq \epsilon^*(\hat{\boldsymbol{\theta}})$. A more complete discussion of measures of robustness can be found in Maronna et al. (2019) and Huber and Ronchetti (2009). To summarize, in this work the main measure of robustness is the finite-sample breakdown point, while also keeping in mind the increase in the MSE or estimation error incurred by contamination, especially in finite-samples.

As has been shown in numerous different settings and applications, the classical LS-estimator of regression, possesses neither of these desired robustness properties (Maronna et al. 2019). Its FBP is $1/n$ and consequently has asymptotic breakdown point of 0; a single aberrant observation can push the estimated regression parameters to infinity. Similarly, the SC and IF are unbounded, and the MB is infinity for any amount of contamination. Under these considerations, a substantial body of research is therefore devoted to find alternatives to the LS-estimator in various settings.

[1] an estimator $\hat{\boldsymbol{\theta}}$ is regression-equivariant if it satisfies $\hat{\boldsymbol{\theta}}(a\mathbf{y}+\mathbf{bX}, \mathbf{CX}) = \mathbf{C}^{-1}(a\hat{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{X})+\mathbf{b})$ for all $a \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^{p+1}$, and all non-singular matrices $\mathbf{C} \in \mathbb{R}^{p+1 \times p+1}$

One prominent approach is to view the LS-loss as the sample variance of the (uncentered) estimated residuals, $s^2(\mathbf{r}) = \sum_{i=1}^n r_i^2/n$. Therefore, the LS-estimator seeks to minimize the sample variance of the residuals. In this light, it seems sensible to replace the sample variance with a robust measure of variability. A measure that is used extensively in univariate scale estimation problems is the median absolute deviation (MAD). In the linear regression context, minimizing the MAD of the residuals is equivalent to minimizing the Least Median Squares (LMS) loss (Hampel 1975; Rousseeuw 1984) given by

$$\mathscr{L}_{\text{LMS}}(\mathbf{y}, \hat{\mathbf{y}}) = \operatorname*{med}_{i=1,\ldots,n} (y_i - \hat{y}_i)^2.$$

The LMS-estimator is consistent for $\boldsymbol{\theta}^0$ and can withstand large amounts of contamination as its finite-sample breakdown point is $\epsilon^*_{\text{LMS}} = \frac{\lfloor n/2 \rfloor - p + 2}{n}$ (Rousseeuw 1984). However, the convergence rate is only of order $n^{-1/3}$ (Kim and Pollard 1990) which implies that in the case of no contamination the LMS-estimator is considerably less efficient than the LS-estimator with a convergence rate of order $n^{-1/2}$. Maybe more problematic for practical considerations, however, is the non-smoothness of the loss function which impedes fast algorithms to compute the estimate.

The issues of the LMS-estimator can be avoided by using a continuous function to define the scale estimator, instead of the median of squared residuals. One such estimator for the residual scale is the M-scale estimator, $\hat{\sigma}_{\text{M}}$ (Huber and Ronchetti 2009), defined as

$$\hat{\sigma}_{\text{M}} \colon \begin{cases} \mathbb{R}^n \to \mathbb{R}_+ \\ \mathbf{r} \mapsto \inf \left\{ s > 0 \colon \dfrac{1}{n} \displaystyle\sum_{i=1}^n \rho\left(\frac{r_i}{s}\right) \leq \delta \right\} \end{cases}. \tag{2.8}$$

This mapping is continuous if the function $\rho \colon \mathbb{R} \to [0, \infty)$ satisfies the condition

[R1] $\rho(0) = 0$ and it is continuous, even, i.e., $\rho(-t) = \rho(t)$, and nondecreasing, i.e., $0 \leq t \leq t'$ implies $\rho(t) \leq \rho(t')$.

Using this M-scale estimate, the corresponding S-estimator (Rousseeuw and Yohai 1984) of linear regression is defined through the S-loss

$$\mathscr{L}_{\text{S}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} \hat{\sigma}^2_{\text{M}}(\mathbf{y} - \hat{\mathbf{y}}). \tag{2.9}$$

As detailed later, the constant δ is essential for the robustness of the S-estimator and needs to satisfy $0 < \delta < \lim_{r \to \infty} \rho(r)$.

The definition of the M-scale estimator, and in turn of the S-estimator, may seem arbitrary, but becomes clearer when considering the equivalent implicit definition

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i}{\hat{\sigma}_{\mathrm{M}}(\mathbf{r})}\right)=\delta,$$

which holds if ρ satisfies condition [R1] and $\hat{\sigma}_{\mathrm{M}}(\mathbf{r})>0$. From the implicit definition and considering the special case $\rho(x)=x^2$, it is evident that in this case $\hat{\sigma}_{\mathrm{M}}(\mathbf{r})=\frac{\delta}{n}\|\mathbf{r}\|_2^2$ and hence the S-estimator coincides with the LS-estimator.

To understand the robustness properties of the S-estimator, it is necessary to first understand them for the M-scale estimator. The M-scale estimator is resistant to grossly aberrant values only if the ρ function is bounded. A bounded ρ function in this work is assumed to satisfy

[R2] $\rho(t)=1$ for all $|t|>c$ with $c<\infty$ and ρ is strictly increasing on $(0,c)$, i.e., $0\leq t<t'<c$ implies $\rho(t)<\rho(t')$.

With a bounded ρ function, the constant δ is in $(0,1)$ and directly affects the robustness of the M-scale estimator with FBP given by $\lfloor n\min(\delta,1-\delta)\rfloor/n$. More specifically, the M-scale estimator with bounded ρ function can tolerate up to $\lfloor n\delta\rfloor$ gross outliers without exploding to infinity, and up to $\lfloor n(1-\delta)\rfloor$ "inliers" without imploding to 0. Because robustness of the M-scale estimator hinges on the boundedness of the ρ function, from now on it is implicitly assumed that the ρ function used for M-scale estimation is bounded.

Independent of the exact choice of the ρ function, as long as it satisfies conditions [R1], [R2] and is continuously differentiable with bounded derivative, the S-estimator is consistent for the true parameters under certain conditions on G_0 (Davies 1990; Smucler 2019). The last condition mentioned for consistency is formalized by

[R3] ρ is continuously differentiable with the derivative $\rho'(t)$ and $t\rho'(t)$ both bounded.

As with the univariate M-scale estimator, the FBP of the S-estimator is determined by δ through $\epsilon_{\mathrm{S}}^*=(\lfloor n\min(\delta,1-\delta)\rfloor-p-1)/n$. Hössjer (1992) show that even with an optimal choice of the ρ function, the efficiency of the S-estimator is inversely proportional to its resistance to outliers; in other words, it cannot be both highly efficient and highly robust. A popular choice for the ρ function that satisfies all of the above conditions is Tukey's

bisquare family of functions given by

$$\rho(t;c) = \min\left(1, 1 - \left(1 - \frac{t^2}{c^2}\right)^3\right) \quad \text{with } c > 0. \tag{2.10}$$

Tukey's bisquare ρ function is convenient to handle in computations and yields an S-estimator which is reasonably close to the S-estimator which uses an "optimal" ρ function in terms of efficiency under the Normal model (Hössjer 1992). It is easy to show that with Tukey's bisquare ρ function, but also many other popular ρ functions, the constant c is merely a scaling factor and does not affect the estimated regression parameters, i.e., $\mathscr{L}_{\mathrm{S}}(\mathbf{y}, \hat{\mathbf{y}}; c) = c^2 \mathscr{L}_{\mathrm{S}}(\mathbf{y}, \hat{\mathbf{y}}; 1)$. In practice, c is usually chosen to yield a consistent estimate of the residual scale under the assumed model G_0, which amounts to $c = 1.5467$ in the case of Gaussian G_0 and a breakdown point of $\delta = 0.5$.

Computation of the S-estimator is challenging because of the non-convexity of the loss function induced by a non-convex ρ function. Consistency and asymptotic Normality of the S-estimator (Rousseeuw and Yohai 1984; Davies 1990; Smucler 2019) only apply to the global minimum of the loss function $\mathscr{L}_{\mathrm{S}}$, which is in practice difficult to find. Optimization algorithms for non-convex problems only converge to a local minimum which depends on the given starting point. Mei et al. (2018) lists several conditions on the ρ function and G_0 under which the loss function has a unique local minimum in an r-ball around the true parameter, i.e., $\{\boldsymbol{\theta} \in \mathbb{R}^{p+1} \colon \|\boldsymbol{\theta} - \boldsymbol{\theta}^0\|_2 < r\}$, with high probability if the sample size $n > Cp\log(p)$. Additionally, this unique local minimum actually corresponds to a global minimum for which the statistical guarantees hold, and gradient descent algorithms converge to it if the starting point is within an $\frac{2r}{3}$-ball of the true parameter. It is therefore necessary to choose the starting points in a strategic way and/or try many different starting points. Although this increases the chance of finding a point within this neighborhood, it is no guarantee.

A key observation to finding good starting points (and to compute the S-estimator) is that the S-loss can be written as a weighted LS-loss,

$$\mathscr{L}_{\mathrm{S}}(\mathbf{y}, \hat{\mathbf{y}}) = \mathscr{L}_{\mathrm{S}}(\mathbf{y}, \mu - \mathbf{X}\boldsymbol{\beta}) = \mathscr{L}_{\mathrm{LS}}(\mathbf{W}_{\boldsymbol{\theta}}\mathbf{y}, \mathbf{W}_{\boldsymbol{\theta}}(\mathbf{1}_n\mu + \mathbf{X}\boldsymbol{\beta}))$$

with a diagonal matrix $\mathbf{W}_{\boldsymbol{\theta}} \in \mathbb{R}^{n\times n}$ of weights that depend on where the loss is evaluated and the data itself. Therefore, the S-estimator also minimizes a weighted LS-loss, where suspicious observations are down-weighted. Because the ρ function is bounded, highly outlying observations can even get 0 weight, which means they are effectively removed from

the equation as if they were not part of the sample.

Based on this observation, a strategy using random subsamples from the data is introduced in Rousseeuw and Leroy (1987) for the LMS- and Least Trimmed Squares (LTS) estimators and optimized for S-estimators in Salibián-Barrera and Yohai (2006) for samples with $n > (p+1)/\delta$. It can be thought of randomly generating the weight matrix $\mathbf{W}_{\hat{\boldsymbol{\theta}}_{\mathrm{S}}}$ by assigning a weight of 1 to a random sample of $p+1$ observations and a weight of 0 to the rest. In essence, the idea is based on the observation that there must be at least one subsample of size $p+1$ which does not contain contaminated observations and the LS-estimator computed on this subsample is close to a global minimum of the S-estimator computed on the complete sample. The justification for this specific size of the subsample is that it must comprise at least $p+1$ observations to ensure a unique solution for the LS-estimator. On the other hand, any subsample greater than $p+1$ is more likely to include contaminated observations. To ensure high probability of actually finding a subsample without contamination, many random subsamples need to be considered. As the size of the subsample grows with the dimensionality, the number of subsamples also needs to increase exponentially with the number of predictors. This makes the strategy unfeasible when p is of moderate size.

A somewhat different strategy is given in Peña and Yohai (1999), who aim to identify possibly influential observations and subsequently compute the LS-estimator without these influential observations. The idea is again that the LS-estimator computed on a "clean" subsample is close to a global minimum of the S-estimator computed on the full sample. Because the strategy by Peña and Yohai uses a more guided scheme to find clean subsamples as compared to random subsampling, the number of subsamples to explore is drastically reduced and only grows linearly with the number of predictors. Another advantage is that the strategy is deterministic and therefore always results in the same S-estimator.

It is important to note that the S-loss has potentially multiple global minima. In particular, the S-loss has a unique global minimum only if every p-dimensional subspace of $(\mathbf{y}, \mathbf{X})$ contains less than $\lfloor n(1-\delta) \rfloor - 1$ observations (Rousseeuw and Yohai 1984; Yohai and Zamar 1988). In other words, if $\lfloor n(1-\delta) \rfloor$ observations can be fit exactly with $\tilde{\boldsymbol{\theta}}$, the S-loss has a global minimum at $\tilde{\boldsymbol{\theta}}$. This is a direct consequence of the smaller effective sample size induced by the bounded ρ function (up to $\lfloor n\delta \rfloor$ observations can have 0 weight). The S-estimator can therefore be sensibly computed only if $p < \lfloor n(1-\delta) \rfloor - 1$, compared to $p < n - 1$ for the LS-estimator.

Regardless of the actual distribution G_0, the S-estimator is highly robust. This high robustness, however, comes at the price of low efficiency under the Normal model compared

to the LS-estimator. Due to this deficiency, the S-estimator is in practice usually only the first step in the multi-tiered MM-estimator (Yohai 1987). The MM-estimator employs the M-loss function defined by

$$\mathscr{L}_{\mathrm{M}}(\mathbf{y}, \hat{\mathbf{y}}; \hat{\sigma}_{\mathrm{S}}) = \frac{1}{n} \sum_{i=1}^{n} \rho_{\mathrm{M}} \left(\frac{y_i - \hat{y}_i}{\hat{\sigma}_{\mathrm{S}}} \right),$$

which quantifies the size of the residuals through a ρ_{M} function satisfying the same conditions as the ρ function for the M-scale, in particular being bounded, and $\rho_{\mathrm{M}}(t) \leq \rho(t)$ for all t. The size of the residuals is taken relative to the scale of the residuals, such that the boundedness of the ρ_{M} function affects only observations with residuals being large relative to the scale of the residuals. This is where the MM-estimator relies on an S-estimate of regression. The scale of the residuals can be estimated from the residuals of the fitted S-estimate, $\hat{\sigma}_{\mathrm{S}} = \hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \hat{\mu}_{\mathrm{S}} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\mathrm{S}})$. The M-loss with bounded ρ_{M} is non-convex and computing MM-estimators in general therefore entails similar challenges as outlined for computing S-estimators. If ρ_{M} and ρ for the initial S-estimate of regression satisfy conditions [R1] and [R2], Yohai (1987) proves that the MM-estimator inherits the breakdown point of the initial scale estimator $\hat{\sigma}_{\mathrm{S}}$, is consistent for $\boldsymbol{\theta}^0$ under mild conditions on the error distribution, and has asymptotic efficiency governed by ρ_{M}. It is therefore possible to increase the efficiency of an MM-estimator without sacrificing robustness.

The bias inflicted by gross errors and efficiency under G_0 depends on the shape of the ρ_{M} function, but more importantly on the cutoff c in condition [R2]. Intuitively, if c is chosen very large, the loss is practically unbounded (and usually behaves like the LS-loss) and gross errors as well as leverage points can damage the estimate. On the other hand, if c is chosen too small, the estimator is inefficient under G_0. Usually, the cutoff c is therefore chosen to yield a certain asymptotic efficiency under G_0 while also limiting the maximum asymptotic bias under contamination. Yohai and Zamar (1997) propose an "optimal" ρ_{M} function in the sense that it is minimizing sensitivity towards contamination while simultaneously achieving a desired asymptotic efficiency.

The MM-estimator is particularly useful in practice because it yields a highly robust and efficient estimate without significantly increasing computational complexity. Even though the M-loss in the second step is non-convex, it is not necessary to find the global minimum of the objective function. Yohai (1987) shows that any local minimum of $\mathscr{L}_{\mathrm{M}}$ close to $\hat{\boldsymbol{\theta}}_{\mathrm{S}}$ has the same asymptotic properties as a global minimum. The practical challenge with MM-estimators, however, is that ρ_{M} needs to be chosen in concordance with ρ and G_0. The

prescribed asymptotic efficiency is achieved by choosing the cutoff c according to the limit of $\hat{\sigma}_{\mathrm{S}}$ under G_0. For these results to be transferable to finite samples, the bias of the M-scale estimate of the residuals must not be too large.

MM-estimators are not the only strategy to compute M-estimators when the residual scale is unknown. Several other estimators augment the objective function to allow for joint estimation of the regression parameters and the residual scale. Options include the concomitant scale estimate (Huber and Ronchetti 2009) and constrained M-estimators (Mendes and Tyler 1996). Usually, these estimators are difficult to compute when using bounded ρ functions because they require certain constraints on the scale to evade global minima at a residual scale of 0. The τ-estimator (Yohai and Zamar 1988) uses a similar strategy through optimization of the τ-loss

$$\mathscr{L}_\tau(\mathbf{y}, \hat{\mathbf{y}}) = \hat{\sigma}^2_{\mathrm{M}}(\mathbf{y} - \hat{\mathbf{y}}) \frac{1}{n} \sum_{i=1}^{n} \rho_\tau \left(\frac{y_i - \hat{y}_i}{\hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \hat{\mathbf{y}})} \right)$$

where ρ_τ is again a bounded ρ function satisfying conditions [R1] – [R3] as well as $2\rho_\tau(r) - r\rho'_\tau(r) \geq 0$. The loss function is very similar to the concomitant scale estimate, but instead of jointly optimizing over the scale and the regression parameters, the scale is given by the M-scale of the residuals. Like the MM-estimator, the τ-estimator can be tuned for high breakdown and asymptotic efficiency. Other robustness-properties (e.g., the maximum bias) are also similar in practice. The main advantage of the MM-estimator over the τ-estimator is that the MM-estimator is easier to compute. Although both the MM-estimator and the τ-estimator can be tuned to have high efficiency, higher efficiency also leads to larger bias under contamination. To keep the bias under contamination reasonably small, a typical choice for the asymptotic efficiency of MM-estimators is 85% in the Normal model. Several one-step procedures to improve upon the asymptotic efficiency as well as the finite sample efficiency of the MM-estimator are discussed in Maronna et al. (2019, Chapter 5.9).

Recently, attention has been directed at circumventing scale estimation for the M-estimator with Huber's ρ function, $\rho(t; c) = \min(t^2/2, c(|t| - c/2))$ by choosing the cutoff value c adaptively. It should be noted that Huber's ρ function is robust towards observations with contamination in the response, but the convex loss function does not protect against influence from aberrant values in the predictors. Loh (2018) constructs a grid of candidate cutoff values $\{3\sigma_{\max} 2^k / 2^K : k = 1, \ldots, K\}$ and chooses the smallest cutoff value such that the difference of the estimate to the estimate at the next larger cutoff is below a certain threshold. Under certain conditions, this procedure leads to consistent parameter estimates

and small bounds on the estimation error. To handle possible leverage points, Loh (2018) suggests a weighting function to down-weight observations with large norm of the predictors. Under the assumption that the error distribution is heavy-tailed (but not contaminated), Sun et al. (2019) choose the cutoff value for the Huber loss $c = k/\sqrt{nlog(n)}\sum_{i=1}^{n}(y_i - \bar{y})^2$ and search for an appropriate multiplier k via cross-validation. The influence of possible leverage points is reduced by univariate winsorizing, i.e., any predictor value larger than a predetermined threshold is replaced by this threshold value. Univariate winsorizing, however, does not take into account the multivariate structure of the data; leverage points are often not overly extreme in a single direction but are extreme when taking into account the overall structure of the predictors. The merit of these works is that they derive non-asymptotic bounds for the L_1 and L_2 estimation error, that hold with high probability under relatively mild conditions. Due to the handling of leverage points, however, neither of these adaptive procedures has a high breakdown point.

The finite sample breakdown point of all robust estimators discussed so far have one key weakness: the breakdown point is lower the closer the number of predictors is to the number of observations. However, not only the robustness properties suffer as the dimension increases, but also the finite-sample and asymptotic efficiency gets worse (e.g., Maronna and Yohai 2010). Albeit much less severe than for robust estimators, the LS-estimator also has higher variability in high-dimensional settings. In the following section, I discuss ways to simultaneously (i) reduce the variability of robust estimators by allowing for a larger finite-sample bias and (ii) make robust estimators applicable to settings where the sample size is less than the number of predictors.

2.3 Estimation Under the Sparsity Assumption

With the surge of data in the last decade, it is increasingly common that the number of potential predictors is in the hundreds or even tens-of-thousands. At the same time, the sample size is only slightly larger than or even smaller than the number of predictors. For example, proteomic technologies measure the expression of hundreds of proteins, but the number of patients in a study is often less than a hundred. In these cases, the estimators introduced in the previous two sections are not well defined. However, by imposing additional restrictions on the true parameter and translating these additional assumptions into constraints on the parameter estimates, the (uncountably) infinite set of global minima of the objective functions for estimators presented in the previous sections can possibly be reduced

to a finite set. In many applications, for instance, it is reasonable to assume that only a few of the many available predictors are actually associated with the response; but it is not known which or exactly how many. In other applications, the number of predictors may not be extraordinarily large compared to the sample size, but the goals of the researcher include to identify the predictors that are actually associated with the response. In both of these scenarios, the assumption can be translated to the linear regression estimation problem by assuming the number of truly relevant, or active, predictors, $\mathscr{A} = \{j \colon \beta_j^0 \neq 0\}$, is much smaller than p. Usually, the size of the active set, $s = |\mathscr{A}|$, is not known. This assumption of sparsity, i.e., only $s \ll p$ predictors have non-zero coefficient, is central to this section and the remainder of this work.

Before discussing ways to leverage the sparsity assumption to estimate the regression parameters in the linear regression model (2.2), I extend the list of desired properties when the sparsity assumption is imposed. Since it is assumed that $p - s$ predictors actually have a coefficient value of 0, it is natural to ask if the predictors with zero coefficient can be recovered with high probability, at least as the sample size increases. This leads to the notion of variable selection consistency. Whereas consistency of the estimator implies that the coefficient approaches its true value, variable selection consistency requires that the probability of all truly inactive coefficients being exactly zero approaches 1, i.e.,

$$\lim_{n\to\infty} \mathbb{P}(\hat{\boldsymbol{\beta}}_{\mathscr{A}^c} = \mathbf{0}_{p-s}) = 1.$$

Here and henceforth, a vector $\boldsymbol{\xi}$ indexed by a set $\mathscr{S}$ (e.g., $\hat{\boldsymbol{\beta}}_{\mathscr{A}^c}$) denotes the vector of elements in $\boldsymbol{\xi}$ with index in the set $\mathscr{S}$, i.e., $\boldsymbol{\xi}_{\mathscr{S}} = (\xi_j)_{j\in\mathscr{S}} \in \mathbb{R}^{|\mathscr{S}|}$. If an estimator is variable selection consistent, one can further ask for the limiting distribution of the parameter estimates for the truly active predictors to be as good as if the true active set would have been known in advance, i.e.,

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{\mathscr{A}} - \boldsymbol{\beta}^0_{\mathscr{A}}) \xrightarrow{d} \mathrm{N}_s(\mathbf{0}_s, \mathbf{V}(\boldsymbol{\beta}^0_{\mathscr{A}})), \tag{2.11}$$

where p, s, and $\mathscr{A}$ possibly grow with n. These two properties together are called the "oracle property" (Fan and Li 2001) as the estimator performs as good as an estimator that knows the true active set (an oracle). Although the oracle property is desired, it will turn out that it is not always easy to obtain an estimator that possesses it; usually this requires several strong conditions on the model and the sample.

The two properties discussed so far in this section are both asymptotic in nature. In the high-dimensional setting it can be desirable to not only consider what happens when

the sample size n increases, but also when the dimensionality p_n is growing with the sample size. In the following, I distinguish between results under fixed dimensionality (i.e., $p_n = p$ remains the same for every sample size), and results under growing dimensionality (i.e., p_n grows with n). Results for growing p_n usually require a condition that p_n does not grow too fast as n tends to infinity, e.g., $\log(p)/n \to 0$ (Bühlmann and van de Geer 2011). Similarly, although not covered in this work, the size of the active set could be allowed to grow with the sample size.

An obvious question is how the LS-estimator performs under the sparsity assumption when the sample size is larger than the number of predictors, $n > p$, and, for example, G_0 is multivariate Normal. Although the LS-estimator is consistent for estimating the parameters, the estimated coefficients of the truly inactive predictors are non-zero with probability 1; only in the limit they are 0. Therefore the LS-estimator does not lead to any variable selection and hence does not fulfill the oracle inequality. The same is true for any of the robust estimators discussed before. It is therefore necessary to look for alternatives with positive probability of setting coefficients actually to 0.

In an idealized world where the number of truly active predictors s is known, a simple strategy for computing an estimator defined by a loss function $\mathscr{L}$ is to determine the subset of predictors of size s which minimizes the loss, i.e.,

$$\underset{\mu\in\mathbb{R},\boldsymbol{\beta}\colon \|\boldsymbol{\beta}\|_0=s}{\arg\min} \mathscr{L}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}). \tag{2.12}$$

This is computationally challenging as the L_0 pseudo-norm $\|\cdot\|_0\colon \mathbf{u} \mapsto \sum_{j=1}^{p} |u_j|^0$ is non-convex and not continuous. A naïve way to find the best subset of size s is to try every single set of s active predictors, which is of course unfeasible unless p and s are small.

If s is unknown, the problem becomes several times more difficult as the minimization problem (2.12) needs to be performed for several (or all p) choices of $0 \leq q = \|\boldsymbol{\beta}\|_0 \leq p$. Furthermore, the obtained solutions for the different choices of q then must be compared using a validation metric to identify the overall best solution. The value of the loss function is an inappropriate metric for comparing the solutions as per definition it decreases for increasing q. Even with recent advances in mixed integer optimization in Bertsimas et al. (2016), which allow more efficient optimization of problem (2.12) over $\boldsymbol{\beta}\colon \|\boldsymbol{\beta}\|_0 \leq q$, it only works for moderately sized problems. Greedy searches, on the other hand, can provide adequate approximations to the best subset regression. Examples include forward stepwise regression, where the search begins with the empty model, $q = 0$, and one predictor at a

time is added such that the loss is minimized among all possible additions. Nevertheless, it is difficult to provide provable statistical guarantees for best subset regression or greedy approximations thereof. Furthermore, Hastie et al. (2017) demonstrate that best subset regression with the LS-loss (and the greedy approximation by forward stepwise regression) often leads to an estimator with small bias but large variance.

Continuous alternatives to the L_0 pseudo-norm are popular tools to improve computational efficiency and decrease the variability of the estimator, usually at the cost of increased bias. Theoretically, any "measure of the size of the coefficient vector", $\Phi\colon \mathbb{R}^p \to [0, \infty)$, can be used to constrain the minimization problem

$$\underset{\mu\in\mathbb{R},\boldsymbol{\beta}\colon\, \Phi(\boldsymbol{\beta})\leq a}{\arg\min} \quad \mathscr{L}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta})$$

to reduce the number of global minima to a finite set if $n > p$. However, a necessary and sufficient condition on Φ for the minimization problem to lead to sparse solutions is that it is nondifferentiable at $\beta_j = 0$, $j = 1, \ldots, p$ (Fan and Li 2001). For convenience, I rephrase the constrained optimization problem in its dual form, which in the context of regression is often called regularized or penalized regression:

$$\underset{\mu\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \mathscr{L}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}) + \lambda\Phi(\boldsymbol{\beta}). \tag{2.13}$$

The hyper-parameter λ is inversely related to the constant a in the constrained optimization problem. If $\lambda = 0$ this is the unregularized minimization problem and identical to (2.6), while $\lambda \to \infty$ necessarily leads to $\hat{\boldsymbol{\beta}} = \mathbf{0}_p$ and thus the estimated active set is empty. Both, the penalty function Φ and the hyper-parameter λ are unrelated to the model and the choice cannot be inferred from the model itself but needs to be done based on external considerations.

Probably the most popular choice for the penalty function for sparse estimation in statistics and beyond is the L_1 norm, $\Phi_1(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. The L_1 norm is the convex envelope of the L_0 pseudo-norm over a small domain and as such yields the closest approximation to best subset regression by means of convex penalty functions (Jojic et al. 2011). When combined with the LS-loss, the L_1 penalty leads to the widely known least absolute selection and shrinkage operator (LASSO) (Tibshirani 1996), henceforth called LS-LASSO to emphasize the specific combination of the loss and penalty function. The LASSO penalty can be motivated from many different angles. Numerous results are available which present different

conditions on the distribution G_0 and the sample under which the LS-LASSO is consistent, variable selection consistent, or possesses the oracle property with growing p_n. Typically, the conditions for the LS-LASSO to have these properties include that the amount of penalization, reflected in λ, vanishes as the sample size increases. While the rate depends on the other conditions imposed, it is usually required to be at most of order $O(\sqrt{\log p_n/n})$. Vanishing regularization is required to remove the bias introduced by the regularization, at least asymptotically. From a practical perspective, this is a relatively mild requirement as with enough data, the LS-estimator will already estimate the coefficients for the truly inactive predictors close to 0 and only a slight nudge is required to make them exactly zero. For a comprehensive summary of conditions and the most important results see Bühlmann and van de Geer (2011).

The elastic net (EN) penalty, proposed by Zou and Hastie (2005) has similar variable selection properties as the LASSO, but is able to retain groups of highly correlated active predictors. The EN penalty is given by a linear combination of the L_1 and squared L_2 penalty,

$$\Phi_{\text{EN}}(\boldsymbol{\beta}; \alpha) = \alpha\|\boldsymbol{\beta}\|_1 + \frac{1-\alpha}{2}\|\boldsymbol{\beta}\|_2^2 \quad \text{with } \alpha \in [0,1]. \tag{2.14}$$

The LASSO is a special case of the EN penalty with $\alpha = 1$, and as long as $\alpha > 0$, the EN penalty has singularities at the origin and therefore also leads to sparse estimates. The L_2 penalty is beneficial in the presence of highly correlated predictors, stabilizing variable selection (Zou and Hastie 2005).

The LS-LASSO and LS-EN estimators possess the oracle property only under very specific and impractical conditions, due to the bias introduced by the L_1 and the L_2 penalty. To solve this problem, a different penalty would need to be considered; one possibility is the family of folded-concave penalties introduced by Fan and Li (2001). Folded-concave penalties are singular at the origin (i.e., produce sparse results) and are bounded, i.e., predictors with coefficients larger than a certain threshold are all penalized equally, regardless of the actual size of the coefficient. The LS-loss combined with folded-concave penalties yields an estimator that possesses the oracle property under growing dimension, requiring less restrictive conditions than the LS-LASSO (Fan and Peng 2004; Zhang and Zhang 2012).

Due to the boundedness of the folded-concave penalties, the objective function (2.13) is non-convex, even if combined with the LS-loss. This proves problematic because the oracle properties and other statistical guarantees are only valid for the global minimum. The local linear approximation (LLA) to folded-concave penalties in combination with the LS-loss is

shown to yield an estimator that has the same properties as the "good" global minimum if $p < n$ (Zou and Li 2008) or if the smallest true coefficient value of the active predictors are large enough and F_0 is sub-Gaussian (Fan et al. 2014).

Fan et al. (2018) improve these results for convex loss functions by introducing a computational framework (I-LAMM) for computing general regularized estimators of the form (2.13), including folded-concave penalties combined with the LS-loss. They cast the estimation problem as an iterative algorithm which, after an infinite number of iterations, coincides with the good global minimum of the LS-loss combined with the folded-concave penalty. However, they also give a bound for the L_2 estimation error that depends on the number of iterations and the chosen numerical accuracy of the obtained solutions. From these bounds it can be seen that the L_2 estimation error approaches the oracle bound if the numerical accuracy is chosen small enough and the number of iterations increases.

Interestingly, the computational framework in Fan et al. (2018) also connects the folded-concave penalties with another important class of penalties: the adaptive LASSO and the adaptive EN. The adaptive EN penalty function (Zou 2006; Zou and Zhang 2009), penalizes the coefficients for each predictor differently depending on the corresponding element in a vector $\boldsymbol{\omega}$ of strictly positive penalty loadings:

$$\Phi_{\text{AN}}(\boldsymbol{\beta}; \boldsymbol{\omega}, \alpha, \zeta) = \frac{1-\alpha}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha \sum_{j=1}^{p} \omega_j^{\zeta} |\beta_j| \quad \text{with } \zeta > 0. \tag{2.15}$$

With the adaptive EN penalty, predictors with a large penalty loading ω_j are more heavily penalized than predictors with a small penalty loading. The penalty loadings are commonly set to the reciprocal values of a preliminary estimate of the regression parameter. Intuitively, if the preliminary estimate is consistent for $\boldsymbol{\beta}^0$, penalization for truly inactive predictors tends to infinity for increasing sample size. Therefore, if the hyper-parameter λ scales appropriately with n, the bias introduced by the penalty becomes negligible and the oracle property can be obtained.

With a slight modification of the penalty loadings, the adaptive LS-LASSO estimator (i.e., $\alpha = 1$) can be obtained after two I-LAMM iterations (Fan et al. 2018). For this equivalence to hold, the weights ω_j must be truncated by $\max(\tau, \omega_j)$ using a reasonably large but finite τ. From this it is easy to obtain bounds for the estimation error of the (modified) adaptive LS-LASSO.

Unsurprisingly, regularized estimators utilizing the LS-loss suffer the same issues as the LS-estimator under contamination, albeit often less obvious. Recalling that the breakdown

point of regression estimators involves the estimated coefficients exploding to infinity, it seems comforting to know that regularized estimators are by definition bounded away from the boundary of the parameter space. In the dual formulation of the regularized loss (2.13), however, the parameter estimate can still diverge to infinity for any fixed $\lambda < \infty$ as shown in Alfons et al. (2013). Furthermore, the intercept parameter μ, is not regularized and can thus also explode under contamination. Even if the model does not include an intercept, the regularization parameter λ poses problems; although λ is not a model parameter and as such is not estimated, the selection of a good λ value is affected by contamination. As shown in Cohen Freue et al. (2019), constraining the slope estimate $\hat{\boldsymbol{\beta}}$ to the interior of the parameter space, λ to would be required to grow indefinitely.

As highlighted by Davies and Gather (2005), the notion of breakdown point is not a sensible measure of robustness for non-equivariant estimators; regularized estimator are per definition not (regression) equivariant. Nevertheless, the breakdown point can still give valuable insights about the robustness properties of an estimator. The maximum MSE under contamination, on the other hand, can be a useful metric for comparing regularized estimators; this is especially true in the presence of leverage points. In increasingly high dimensions it is more important to have estimators that are insensitive to leverage points. Although Huber and Ronchetti (2009) suggest identifying possible leverage points in advance and analyze them separately, in high dimensional problems this approach is impractical because leverage points are very difficult to identify. Even if it is possible to identify leverage points, under the sparsity assumption, it is not sensible to take aside observations with potential leverage coming from the truly inactive predictors. However, because it is unknown which predictors are truly active and inactive it is impossible to "screen out" leverage points for separate analysis prior to computing an estimate.

Just as without the sparsity assumption it is therefore necessary to devise methods which can achieve low MSE but additionally identify important predictors even under arbitrary contamination. Because the maximum MSE under contamination is usually impossible to derive theoretically, it is also still desirable to achieve a high breakdown point, even though it is not the best measure of robustness for regularized estimators.

2.4 Robust Regularized Estimation

The main culprit in the erratic behavior of regularized estimators under contamination is still the LS-loss. Drawing from the insights gained in unregularized estimation, it therefore

seems sensible to replace the LS-loss with a robust surrogate.

Due to its importance for quantile regression, the LAD-LASSO (Wang and Li 2007) is among the first regularized regression estimators with robustness towards gross errors. Numerous papers study the behavior of M-loss functions with convex ρ functions (e.g., Huber's ρ) combined with the L_1 penalty under different settings. Many of the properties of the LS-LASSO also hold for convex M-estimators under similar conditions (van de Geer and Müller 2012). Recently, several strategies to reduce the bias introduced by the convex M-loss as well as to avoid residual scale estimation have been proposed (Loh 2018; Fan et al. 2016; Fan et al. 2017; Fan et al. 2018; Sun et al. 2019; Yang 2017).

Robust regularized estimation is not the only strategy for robust estimation in the sparse linear regression model. Khan et al. (2007), for example, propose the Robust Least Angle Regression (RLARS) estimator to compute robust regression estimates in a step-wise manner. Following ideas of the LARS estimator (Efron et al. 2004), the steps are taken in the direction of the predictor with highest correlation with the residuals from the previous step. RLARS gains robustness towards arbitrary contamination by using robust measures of location, scale, and correlation for selecting and taking the steps. Empirical results suggest RLARS is reliable under gross contamination, but the finite-sample bias is often higher than of other robust methods and the algorithmic definition of RLARS hinders the establishment of theoretical guarantees.

Given the increased difficulties caused by leverage points, the bounded M-loss is an indispensable tool in higher dimensions. However, considerably less attention has been given to LASSO-type M-estimators with non-convex or bounded ρ functions as well as S-estimators. Smucler and Yohai (2017) proves that the MM-LASSO, the estimator that minimizes a redescending M-loss combined with the LASSO penalty, is $\sqrt{n}$-consistent for $\boldsymbol{\theta}^0$ when the dimension is fixed but otherwise very mild conditions. Importantly, there are no moment-conditions on F_0; the errors only need to have a density that is symmetric around 0 and monotonically decreasing in $|u|$ and strictly decreasing in a neighborhood of 0. An additional condition is that the second moment of H_0 must be finite, and the covariance matrix of the predictors needs to be non-singular. Therefore, the MM-LASSO is consistent even under very heavy-tailed distributions F_0, such as the Cauchy distribution. Similar results, albeit under more restrictive assumptions, specifically finite second moment of the error F_0, are obtained in Arslan (2016) and Chang et al. (2018).

Loh (2017) studies the finite-sample bounds of the L_1 and L_2 estimation errors of redescending regularized M-estimators, including those with a LASSO penalty. She shows

that any minimum of the objective function (not only a global minimum), which lies in an r-ball around the true parameter, fulfills the oracle inequality for the estimation error. As can be expected of finite-sample results for complicated non-convex estimation problems, there are several technical conditions for this result to hold:

1. The regularized objective (2.13) is restricted to an R-box around the origin, $\{\boldsymbol{\beta}\colon \|\boldsymbol{\beta}\|_1 < R\}$ which needs to contain the true parameter, i.e., $\|\boldsymbol{\beta}^0\|_1 < R$, and as such requires to have a rough idea of the size of the true parameter; the larger R, the weaker the bound on the estimation error.

2. The sample size needs to be large enough to guarantee that there is at least one minimum in an r-ball around the true parameter with high probability; the smaller r the larger a sample size is needed. With an even larger sample size, every minimum in the R-box around the origin falls within the r-ball around the true parameter with high probability.

3. The gradient of the M-loss evaluated at the true parameter needs to be bounded with high probability.

4. Most importantly, the M-loss needs to satisfy the restricted strong convexity (RSC) condition in an r-ball around the true parameter with high probability. This condition is essentially bounding the "non-convexity" of the loss function $\mathscr{L}$ around the true parameter; the more non-convex the larger the bound on the estimation error.

Establishing these conditions is difficult in theory for a given G_0 and almost impossible in practice. To overcome these difficulties, Loh (2017) states different sufficient conditions on G_0 under which the above conditions hold with high probability. For example, the gradient is bounded with high probability if the distribution of the predictors, H_0, is sub-Gaussian, i.e., has lighter tails than a multivariate Normal distribution. Furthermore, under sub-Gaussian predictors and a specific tail-behavior of the errors F_0, the RSC condition also holds with high probability.

We recently proposed the first S-estimator with an elastic net penalty (Cohen Freue et al. 2019) called Penalized Elastic Net S-Estimator (PENSE) which shares many of the properties of the MM-LASSO, without the need for an auxiliary scale estimate. Chapter 3 gives a detailed exposition of the EN penalty, its advantages over the LASSO, and the theoretical properties and empirical results concerning PENSE. Importantly, PENSE has

very good robustness properties and is root-n consistent for the true regression parameter under fixed dimension.

The only other regularized S-estimator proposed so far is the S-Ridge (Maronna 2011). The S-Ridge combines the S-loss with the Ridge penalty, i.e., the squared L_2 norm of the coefficients (also a special case of PENSE with $\alpha = 0$). The Ridge penalty does not induce sparsity, i.e., none of the estimated coefficients will be 0, but it helps in high-dimensional problems to reduce the variability of the estimate at the cost of increased bias. Smucler and Yohai (2017) prove that the S-Ridge is a consistent estimator for the true regression parameter and the residual scale. This allows them to use the S-Ridge estimator to obtain an auxiliary estimate of the residual scale for their MM-LASSO estimator. Despite the different penalties involved, the authors also use the S-Ridge estimate as the starting point for the optimization of the non-convex MM-LASSO objective function. Although there is no guarantee that this will yield a sensible estimate, the empirical performance of the MM-LASSO is very competitive.

There also exist results for M-estimators with different penalty functions. The theory in Loh (2017) for M-estimators covers folded-concave penalties and the results establish the oracle property for a broad class of loss functions, given that the above-mentioned conditions (and a stronger RSC condition) hold with high probability. Fan et al. (2018) establish the error-bounds for estimates computed by the I-LAMM procedure with high probability for the LS-loss and a sub-Gaussian G_0, but their theory also allows for different convex loss functions. It remains open, however, if the conditions for their results can be obtained with high probability when using non-convex, redescending M-estimators under heavy-tailed errors and contamination in the predictors.

In Cohen Freue et al. (2019) we propose a refinement step to PENSE, called PENSEM. The idea of PENSEM is similar to MM-estimators for low-dimensional regression, improving efficiency by a subsequent M-step which relies on a scale estimate obtained from the residuals of the fitted PENSE estimate. This refinement works well in many problems, but, as detailed in Chapter 5, the residual scale estimate from PENSE and other robust estimators can be very biased in high-dimensional problems. In finite samples, this bias may impede gains in efficiency and make the M-step possibly susceptible to contamination.

The adaptive MM-LASSO (Smucler and Yohai 2017; Chang et al. 2018) combines a bounded M-loss with the adaptive LASSO penalty, and it is shown in Smucler and Yohai (2017) that this estimator possess the oracle property under the same mild conditions as for root-n consistency of the MM-LASSO. To avoid the necessity of an initial scale estimate,

I introduce the adaptive PENSE in Chapter 3. The adaptive PENSE combines the S-loss with the adaptive EN penalty and uses PENSE as preliminary estimate. I show that the adaptive PENSE also possess the oracle property under the same conditions as needed for PENSE to be root-n consistent.

Chapter 3

Elastic Net S-Estimators

This chapter introduces a novel estimator for the linear regression model under the sparsity assumption which can tolerate the presence of a large proportion of adverse contamination. The challenge of obtaining a robust estimate of the residual scale under the sparsity assumption, especially in high dimensional problems, hampers the application of regularized M-estimators. In Cohen Freue et al. (2019), we therefore propose the penalized elastic-net S-estimator (PENSE), which combines the robust S-loss function with an elastic net penalty. PENSE circumvents the need of an auxiliary scale estimate.

3.1 Method

The PENSE estimator is defined by a regularized objective function which combines the classical S-loss (2.9) and the EN penalty (2.14):

$$\mathscr{O}_{\text{S}}(\mu, \boldsymbol{\beta}; \lambda, \alpha) = \mathscr{L}_{\text{S}}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}) + \lambda \Phi_{\text{EN}}(\boldsymbol{\beta}; \alpha). \tag{3.1}$$

Minimizers of this objective function are denoted by $\tilde{\boldsymbol{\theta}}^{(\lambda,\alpha)} = \arg\min_{\mu,\boldsymbol{\beta}} \mathscr{O}_{\text{S}}(\mu, \boldsymbol{\beta}; \lambda, \alpha)$, while the arguments λ or α are omitted if irrelevant or obvious from the context.

Due to the non-convexity of the S-loss, the PENSE objective function is also non-convex. Without non-convexity, PENSE would not possess its robustness properties as detailed in Section 3.4, but it is also the source of computational challenges. The issue is not unique to PENSE but is shared among all S- and redescending M-estimators with and without regularization. The robustness properties and statistical guarantees for the non-regularized S-estimator only pertain to a global minimum (Davies 1990; Smucler 2019) of the S-loss.

The asymptotic statistical properties of PENSE detailed in Section 3.3 also only pertain to the global minimum and are contingent on λ decreasing fast enough. In other words, λ cannot be too large for the global minimum to have good statistical properties. This is in line with conditions for asymptotic properties of LS-EN and LS-LASSO estimators, albeit their objective functions are convex and a large regularization parameter merely introduces too much bias to attain a minimum with provable properties. For PENSE, on the other hand, too large λ values not only introduce bias but also abets the estimator's robustness. For large λ, local minima of the objective function that are close to the origin are more likely also global minima. Hence, if λ is too large global minima could very well be artifacts of contamination and not sensible estimates. One such instance is depicted in Figure 3.1 for a simple regression model without intercept and a single predictor. The true regression coefficient is $\beta^0 = 1$, but for $\lambda = 1$ the objective function exhibits a global minimum around $\beta = -0.5$ due to contamination in the sample. Only as λ gets smaller, the "good" minimum around $\beta = 1$ becomes a global minimum. A sensible PENSE estimate can therefore be attained only if an appropriate strategy to select the regularization parameter λ is used.

Although only global minima have provable statistical properties, for larger λ values, local minima not caused by contamination can still be useful to predict the expected value of the response, given a set of predictor values. Prediction, alongside identifying the predictors important to make good predictions, is a main goal in many applications of regularized estimators. Therefore, it is important to not only check the global minima for their predictive capability, but also other local minima, even though they might not possess the same statistical properties as the global minima.

As noted in Chapter 2, the S-loss and therefore the PENSE objective function can be rewritten as a weighted LS-EN objective function

$$\mathscr{O}_{\mathrm{S}}(\mu, \boldsymbol{\beta}; \lambda, \alpha) = \frac{1}{2n} \sum_{i=1}^{n} w_i^2(\mathbf{r}) r_i^2 + \lambda \Phi_{\mathrm{EN}}(\boldsymbol{\beta}; \alpha) =: \mathscr{O}_{\mathrm{EN}}(\mu, \boldsymbol{\beta}; \mathbf{w}(\mathbf{r}), \lambda, \alpha) \tag{3.2}$$

with residuals $r_i = y_i - \mu - \mathbf{x}_i^\intercal \boldsymbol{\beta}$ and weights

$$w_i(\mathbf{r}) = \hat{\sigma}_{\mathrm{M}}(\mathbf{r}) \sqrt{\frac{\rho'\left(\frac{r_i}{\hat{\sigma}_{\mathrm{M}}(\mathbf{r})}\right)/r_i}{\frac{1}{n}\sum_{k=1}^{n} \rho'\left(\frac{r_k}{\hat{\sigma}_{\mathrm{M}}(\mathbf{r})}\right) r_k}}. \tag{3.3}$$

This representation of the PENSE objective function allows for an intuitive interpretation of the estimator. The PENSE estimate corresponds to a properly weighted LS-EN estimate,

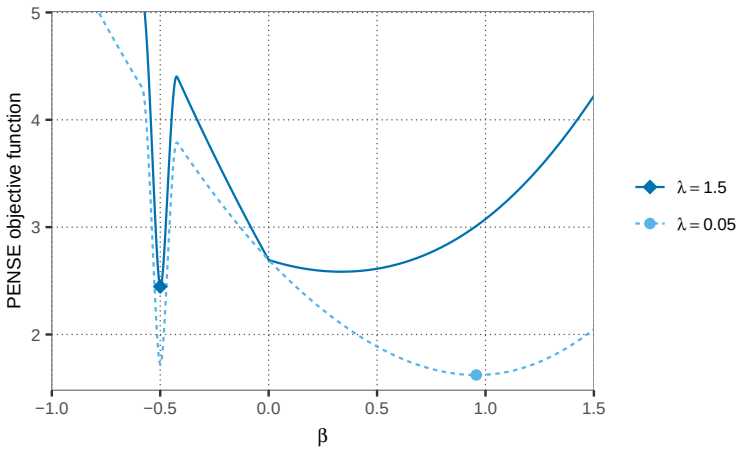


Figure 3.1: PENSE objective function (3.1) for a simple linear regression model of the form $\mathbf{y} = \mathbf{x} + \mathbf{u}$, evaluated at different values of β and λ on a data set with contamination. The marked dots depict the locations of the global minima for different λ.

where the weights are chosen to down-weight the contaminated observations and to give more weight to proper observations.

The challenges in computing and applying the PENSE estimate are (i) to find global minima of the objective function and (ii) choose a regularization parameter λ such that the global minima enjoy good statistical properties. Additionally, it is also advisable to retain other local minima and determine their predictive abilities. Numerical algorithms to find stationary points of (3.1) require a starting point as input and typically converge to a stationary point which depends on this starting point. To find global minima of the objective function, it is therefore necessary to have starting points that are close to global minima. Local minima are caused by contamination and unusually large error terms, and hence a sensible strategy to find starting points is to compute a LS-EN estimate on a subset of the data which does not contain observations exerting high leverage on the estimate. This direct relationship between the data and the presence/location of local minima is a clear advantage over non-convexity caused by folded-concave penalties. With folded-concave penalties, local minima are due to the underlying regression parameters and no intuitive strategy is available which is known to give starting points close to the desired optimum. Under a restricted eigenvalue condition, sub-Gaussian errors and large enough true coefficient values, Fan et al. (2014) show that with high probability the LS-LASSO is a starting point for their algorithm which leads to the desired local minimum. Although the intuition behind good starting points for optimizing the PENSE objective function is simpler and holds without any restrictive conditions, it is nevertheless challenging to

determine good subsets of the data.

3.2 Initial Estimator

This section discusses different strategies to obtain starting values for locating minima of the PENSE objective function. As outlined above, the landscape of the objective function is scattered with local minima and the goal is to find local and global minima that are not caused by contamination.

3.2.1 Random Subsampling

The most common strategy to determine initial estimates for unregularized S-estimators as proposed in Rousseeuw and Yohai (1984) and Salibián-Barrera and Yohai (2006) is to randomly select subsets of the available observations and compute the classical LS-estimate using only the random subset. The motivation behind the strategy is to get a crude approximation to the weights (3.3) at a global minimum. In the unregularized case, to guarantee that the resulting S-estimator has a breakdown point of ϵ with probability at least ν, the lower bound for the number of subsets N is given by

$$N \geq \frac{\log(1-\nu)}{\log(1-(1-\epsilon)^{p+1})}$$

and thus grows exponentially with p (Salibián-Barrera and Yohai 2006). With N subsets, the probability that at least one of the subsamples of size $p+1$ is "clean", i.e., does not contain any contaminated observations is ν. However, even an initial estimator computed on such a "clean" subsample does not necessarily lead to a global optimum of the S-estimator. Hence, it is in general not enough to examine a single clean subset, increasing the required number of subsamples even further. While Salibián-Barrera and Yohai (2006) propose several computational shortcuts to make random subsampling feasible for the unregularized S-estimator with up to a moderate number of predictors, in higher dimensional settings the computational burden of finding a global minimum with high probability is insurmountable.

Random subsampling is similarly used for robust regularized estimation, where the penalty term could potentially reduce computational challenges, even in high dimensions. Due to regularization, the size of the subset can be much smaller than the number of predictors. Alfons et al. (2013), for example, use random subsets of size 3 to obtain initial estimates for their SparseLTS estimator. By decoupling the size of the subset from the

number of predictors, the number of subsets required to get at least one clean subsample with high probability only increases exponentially with the chosen size of the subset. Although this implies that only a few subsets are required, subsamples of very small size (e.g., 3 as for SparseLTS) correspond to an approximation of the weights (3.3) at a global minimum of the PENSE objective function by a vector with only 3 non-zero entries, which is likely inaccurate considering that the vector of weights at a global minimum has at least $\lfloor(1-\delta)n\rfloor$ non-zero entries. Therefore, to maintain a high likelihood of locating a global minimum (or a good local minimum) it is still necessary to consider a very large number of random subsamples; either because clean subsamples of small size are likely not a good initial estimate, or because a large subsample likely contains contamination. This is a major obstacle for using random subsampling to initialize PENSE or other robust regularized estimators.

3.2.2 Elastic Net Peña-Yohai Procedure

The problem with random subsampling, i.e., the large number of subsets required to increase the chance of finding a good local optimum, stems from the fact that the subsets are chosen without considering the data itself. The following strategy, proposed by Peña and Yohai (1999) as outlier detection method and standalone estimator for linear regression, on the other hand, aims at identifying and omitting contaminated observations. The Peña-Yohai (PY) procedure builds several subsets of the data, each of which omits observations with possibly large influence on the LS-estimate, computes the LS-estimate for each of these subsets, and finally chooses the estimate whose residuals have the smallest M-scale. The PY procedure mainly screens out observations with high leverage, while retaining observations with small leverage but large residuals. To remove the influence of these observations as well, the PY procedure is iterated several times by removing the observations with large residuals in the fit with the smallest M-scale of the residuals. Although Peña and Yohai (1999) propose their procedure for the unpenalized S-estimator, Maronna (2011) successfully adapts the PY procedure to find initial estimates for the S-Ridge estimator. In Cohen Freue et al. (2019), we adapt the PY procedure for general, non-linear, regularized estimators such as PENSE, by employing regularized LS-estimators throughout the procedure. The PY procedure adapted for regularized estimation using the EN penalty (EN-PY) is outlined in Algorithm 1 for fixed penalty parameters λ and α.

The central piece in the EN-PY procedure is the set of possibly clean subsets, line 5 in Algorithm 1. Peña and Yohai (1999) derive this set using the principal sensitivity compo-

Algorithm 1 EN-PY Procedure

Input: Fixed penalty parameters λ and α, the proportion of observations in each clean subset, κ, $0 < \kappa < 1$, a cutoff value $C > 0$ for "large" residuals, and the maximum number of PY iterations I.

1: Initialize the set of indices with the full data set, $\mathscr{I}^{(0)} = \{1, \ldots, n\}$.
2: Set $\iota = 0$.
3: **repeat**
4: Compute the LS-EN estimate for fixed λ, α with all observations in the current index set $\mathscr{I}^{(\iota)}$, $\tilde{\boldsymbol{\theta}}^{(0)}$.
5: Obtain a set of possibly clean subsets of $\mathscr{I}^{(\iota)}$, $\{\mathscr{S}_1, \ldots, \mathscr{S}_K\}$, each of size $\lfloor \kappa |\mathscr{I}^{(\iota)}| \rfloor$ and $\mathscr{S}_k \subset \mathscr{I}^{(\iota)}$.
6: **for** $k = 1, \ldots, K$ **do**
7: Compute the LS-EN estimate for fixed λ, α on the subset $\mathscr{S}_k$, $\tilde{\boldsymbol{\theta}}^{(k)}$.
8: **end for**
9: Choose the LS-EN estimate that results in the smallest M-scale of all n residuals,

$$\hat{\boldsymbol{\theta}}^{(\iota)} = \tilde{\boldsymbol{\theta}}^{(k')} \quad \text{with } k' = \underset{k=0,\ldots,K}{\arg\min}\, \hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \mu^{(k)} - \mathbf{X}\tilde{\boldsymbol{\beta}}^{(k)}).$$

10: Update the index set to include only observations with small standardized residuals,

$$\mathscr{I}^{(\iota+1)} = \left\{ i = 1, \ldots, n \colon \left| y_i - \hat{\mu}^{(\iota)} - \mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}^{(\iota)} \right| < C \hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \mu^{(\iota)} - \mathbf{X}\tilde{\boldsymbol{\beta}}^{(\iota)}) \right\}.$$

11: Increment ι, $\iota = \iota + 1$.
12: **until** $\iota = I$ or the index set did not change, $\mathscr{I}^{(\iota)} = \mathscr{I}^{(\iota-1)}$
13: **return** all $K+1$ estimates $\left\{ \tilde{\boldsymbol{\theta}}^{(k)} \colon k = 0, \ldots, K \right\}$ from the last EN-PY iteration, $\iota - 1$.

nents (PSCs); a set of directions in which points of high leverage should appear as large values. For EN-PY, the principal sensitivity components are obtained from the $n \times n$ matrix of leave-one-out (LOO) residuals, $\mathbf{R}$; the k-th column of $\mathbf{R}$ is the vector of differences between the observed $\mathbf{y}$ and the values fitted by an LS-EN-estimate computed from all but the k-th observation (line 2 in Algorithm 2). The PSCs are defined as the projections of matrix $\mathbf{R}$ on its eigenvectors. It can be shown (Peña and Yohai 1999) that observations with very high leverage have an extreme value (positive or negative) in at least one PSC. From each PSC, three subsets of size m are obtained from: (a) the m observations with smallest values in this direction (i.e., filter extremely positive values), (b) the m observations with largest values (i.e., filter extremely small values), and (c) the m observations with smallest absolute values (i.e., filter extremely positive or negative values). The detailed procedure to derive subsets from the PSCs for PENSE is given in Algorithm 2.

The Peña-Yohai procedure for regularized estimators as detailed in Algorithms 1 and 2

Algorithm 2 Subsets derived from the Principal Sensitivity Components

Input: Fixed penalty parameters λ and α, an index set $\mathscr{I}$ of cardinality $\tilde{n}$ and the desired proportion of indices in each subset, $\kappa < 1$.

1: Define the desired size of the subsets as $m = \lfloor \kappa \tilde{n} \rfloor$.

2: Compute the $\tilde{n} \times \tilde{n}$ sensitivity matrix $\mathbf{R}$. The entries of $\mathbf{R}$ are given by

$$R_{i,k} = y_i - \hat{\mu}_{(-k)} - \mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}_{(-k)} \qquad i,k = 1,\dots,\tilde{n},$$

where $\hat{\boldsymbol{\theta}}_{(-k)}$ is the LS-EN estimate computed for fixed λ, α, from the observations in the index set $\mathscr{I}$ with the k-th entry omitted, i.e., the leave-one-out LS-EN estimate.

3: Determine Q, the number of non-zero eigenvalues of the matrix $\mathbf{R}^\intercal\mathbf{R}$.

4: **for** $q = 1,\dots,Q$ **do**

5: Compute the q-th PSC, $\mathbf{z}^{(q)} = \mathbf{R}\mathbf{v}^{(q)}$, where $\mathbf{v}^{(q)}$ is the q-th eigenvector of $\mathbf{R}^\intercal\mathbf{R}$.

6: Define the subset with the m observations with smallest values in $\mathbf{z}^{(q)}$, i.e.,

$$\mathscr{S}_q = \{i = 1,\dots\tilde{n} \colon z_i^{(q)} < Z_s\} \text{ with } Z_s = \inf\left\{ Z \colon m \leq \sum_{i=1}^{\tilde{n}} \mathbb{I}\left\{ z_i^{(q)} < Z \right\} \right\}$$

7: Define the subset with the m observations with largest values in $\mathbf{z}^{(q)}$, i.e.,

$$\mathscr{S}_{Q+q} = \{i = 1,\dots\tilde{n} \colon z_i^{(q)} > Z_l\} \text{ with } Z_l = \sup\left\{ Z \colon m \leq \sum_{i=1}^{\tilde{n}} \mathbb{I}\left\{ z_i^{(q)} > Z \right\} \right\}$$

8: Define the subset with the m observations with smallest absolute values in $\mathbf{z}^{(q)}$, i.e.,

$$\mathscr{S}_{2Q+q} = \{i = 1,\dots\tilde{n} \colon |z_i^{(q)}| < Z_a\} \text{ with } Z_a = \inf\left\{ Z \colon m \leq \sum_{i=1}^{\tilde{n}} \mathbb{I}\left\{ |z_i^{(q)}| < Z \right\} \right\}$$

9: **end for**

10: **return** the set $\{\mathscr{S}_1,\dots,\mathscr{S}_{3Q}\}$.

generates a total of $3Q+1$ initial estimates for computing the PENSE estimate. The major benefit of EN-PY over random subsampling is that $Q \leq \max(p,n)$, and hence the number of initial estimates from the EN-PY procedure only grows linearly with the number of observations and the number of predictors, as opposed to the exponential growth required for random subsampling. Therefore, by choosing the subsets in a more guided fashion, the computation can be greatly reduced compared to naïve subsampling.

For the case of unpenalized regression, Peña and Yohai (1999) present several mathematical shortcuts to efficiently derive the PSCs. These shortcuts are based on the closed form solution for LOO residuals in the case of linear estimators and thus cover the ordinary LS-estimator as well as the LS-Ridge estimator. Unfortunately, there is no counterpart of

these closed form solutions for regularized estimators with non-smooth penalty function. Therefore, the bottleneck of the EN-PY procedure is the cumbersome computation of the LOO residuals. For a fixed value of λ and α, the EN-PY procedure requires the computation of at most $n(4 + 4I + I\kappa) + I + 1$ LS-EN estimates, where I and κ as in Algorithm 1. Nevertheless, the actual number of LS-EN estimates that need to be computed is usually much smaller than this upper bound since the residual-filtered index set most often remains constant after a few iterations. Hence, even without mathematical shortcuts to compute the PSCs, the EN-PY procedure is still significantly faster than random subsampling for obtaining initial estimates for PENSE.

In addition to the PY procedure, Peña and Yohai (1999) propose an estimate for linear regression based on a one-step re-weighting of the S-estimate obtained from the "best" estimate (in terms of minimal M-scale of the residuals of all observations) computed through the PY procedure. Their estimate is a weighted LS estimate, where hard-rejection weights (0/1) are derived from the residuals of this aforementioned S-estimate. The weights, however, are derived from the "hat" matrix of their linear estimator and the idea is therefore not transferable to regularized estimates.

3.2.3 Empirical Comparisons

The main selling point for the EN-PY procedure is the decreased computational burden by selecting the subsets in a way that excludes potentially contaminated observations. With the same number of initial estimates, the chance that the EN-PY procedure gives at least one good initial estimate should be higher than with random subsampling. Peña and Yohai (1999) show that high leverage points are detectible in at least on PSC direction. The authors claim that due to this property the PY procedure can efficiently clean the data of gross contamination. Although the theory presented in the paper does not cover moderate leverage points, the results of simulation studies further underline the benefits of the PY procedure.

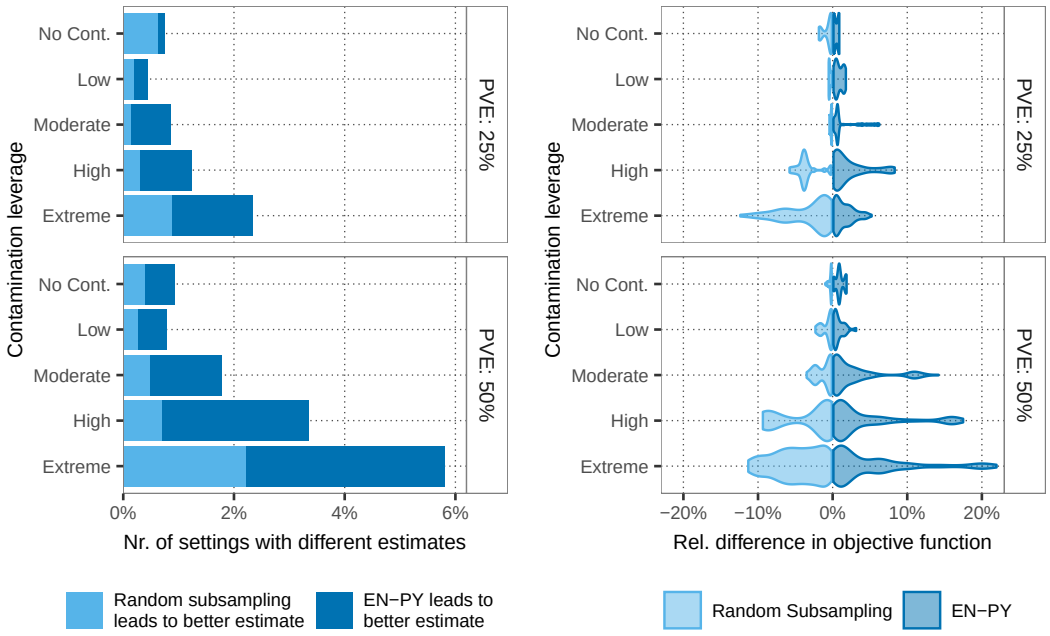
To ascertain that the advantages of the PY procedure translate to similar properties of the EN-PY procedure for sparse linear regression, I compare EN-PY and random subsampling empirically. For this experiment, data sets with $n = 100$ observations and $p = 16$ predictors are randomly generated according to 42 scenarios following scheme *VS1-LT** (see Appendix A.1.1). In this lower-dimensional problem the likelihood of uncovering at least one clean subset with a computationally feasible number of random subsamples is still high. The scenarios are divided into two groups: two scenarios where no contamination is intro-

duced and 40 scenarios where 25% of the observations are contaminated. In scenarios with contamination, the placement of contaminated observations is controlled by the leverage of contaminated observations as well as the regression parameter in the linear model generating these contaminated observations. The variance of the error term is chosen such that the percentage of variance explained (PVE) by the true regression model is either 25% or 50% (this amounts to a signal-to-noise ratio of 1/3 and 1, respectively, and follows the suggestions in Hastie et al. (2017)). Appendix A.2 gives the complete details of the scenarios considered in this numerical experiment.

For each generated data set, initial estimates are obtained at 10 different penalization levels using random subsampling and the EN-PY procedure. All initial estimates from different penalization levels are merged into two sets: $\mathscr{T}_{\text{RS}}$ comprising initial estimates from random subsampling and $\mathscr{T}_{\text{EN-PY}}$ for initial estimates from the EN-PY procedure. The PENSE estimate is then computed for 50 different values of the penalization level. At every penalization level, the PENSE estimate is computed once from initial estimates $\mathscr{T}_{\text{RS}}$ and once from initial estimates $\mathscr{T}_{\text{EN-PY}}$, recording the difference in the attained value of the objective function.

The main results of this experiment are depicted in Figure 3.2. The left plot shows the number of settings (i.e., combinations of data sets and penalization levels) where a difference between the EN-PY and random subsampling procedures is detected. For the vast majority of settings, both procedures lead to the same minimum being uncovered, but more severe leverage points lead to more differences between the two procedures. This can be expected because the PENSE objective function usually exhibits more local optima the more severe leverage points are present. Of those replications where EN-PY and random subsampling lead to different local optima, the local optimum uncovered by EN-PY is most often better than the local optimum found via random subsampling. Interestingly, the differences are more plentiful when the variance of the error term is small (PVE of 50%).

For those replications where there is a difference between the two procedures, the right plot (Figure 3.2(b)) shows the magnitude of these differences, relative to the true variance of the error term. Although EN-PY does not always lead to better optima, if there is a difference, the local optimum uncovered by EN-PY initial estimates are sometimes substantially better than the local optimum obtained from random subsampling. Relative to the true variance of the error, EN-PY sometimes leads to a local optimum more than 20% better than the local optimum attained from starting at initial estimates obtained by random subsampling.



(a) Number of estimates with different objective value.

(b) Magnitude of differences.

Figure 3.2: Comparison of the PENSE objective function at the best minimum uncovered by the EN-PY initial estimates and random subsampling initial estimates. Plot (a) shows the number of settings (relative to the total number of combinations of data sets and penalization levels considered in each scenario) where either EN-PY or random subsampling resulted in a lower value of the objective function. Plot (b) shows the actual difference (relative to the true variance of the errors) between the local optima obtained through EN-PY initial estimates and random subsampling. Positive values indicate the EN-PY initial estimate resulted in a smaller value of the PENSE objective function.

Overall, there seem to be only small differences between initial estimates obtained through random subsampling and EN-PY. These small differences, however, suggest favoring EN-PY for many configurations of the data. To compare computational complexity of the two procedures in this experiment, the number of initial estimates obtained via random subsampling is set to the number of LS-EN estimates computed for EN-PY. While the number of LS-EN problems is the same, the similarity of LS-EN problems involved in the EN-PY procedure makes it on average 2.9 times as fast as random subsampling for computing the initial estimates alone. Even more importantly, EN-PY leads to a much smaller number of initial estimates that have to be considered when computing the PENSE estimate. Given that the computation of the PENSE estimate for each initial estimate is computationally challenging even when using optimizations, the savings in computation time when using EN-PY are substantial. In this experiment, it takes on average 8.7 times

longer to compute PENSE estimates using initial estimates from random subsamples then when using initial estimates from EN-PY. The quality of local minima uncovered by EN-PY is better than those uncovered by random subsampling, yet computing PENSE estimates using EN-PY is several times faster, suggesting EN-PY initial estimates are highly preferable.

3.2.4 Initial Estimates for a Set of Penalization Levels

Random subsampling and the EN-PY procedure produce initial estimates for a fixed penalization level. In practice, however, a good penalization level is unknown in advance and PENSE must be computed for an entire set of penalization levels. The number of selected variables and prediction performance of the estimate vary greatly among different penalization levels; hence a fine grid of many penalty levels is preferred. Computing initial estimates for every value in this large set of penalty levels, $\mathscr{Q}$, is infeasible. The fine granularity of $\mathscr{Q}$, on the other hand, allows for an efficient strategy of "warm-starts" as devised in Cohen Freue et al. (2019).

Consider a grid $\mathscr{Q}$ containing $Q > 1$ penalization levels in descending order, i.e., $\mathscr{Q} = \{\lambda_1, \ldots, \lambda_Q\}$ such that $\lambda_{q-1} > \lambda_q$ for $q = 2, \ldots, Q$. Further, denote by $\hat{\boldsymbol{\theta}}^{(q-1)}$ a local minimum of the PENSE objective function at λ_{q-1}. Since the grid is fine-grained, λ_{q-1} and λ_q are not too far apart, suggesting a local minimum of the objective function at λ_q is likely close to $\hat{\boldsymbol{\theta}}^{(q-1)}$. If more than one local minimum at λ_{q-1} is uncovered, each of these minima can be used as initial estimate at λ_q. These warm-starts are repeated at each $\lambda \in \mathscr{Q}$, thereby "following" local minima over different penalization levels. As depicted in Figure 3.1, this strategy can greatly increase chances of uncovering global minima as a local minimum may transmute to a global minimum as the level of penalization changes.

The warm-starts of course depend on local minima uncovered at the preceding penalization level. Therefore, at some point, a different approach for computing initial estimates is necessary. The simplest form is the "0-based" regularization path. For a large enough penalization level, the 0-vector, $\boldsymbol{\beta} = \mathbf{0}_p$ is a local minimum of the PENSE objective function and thus can be traced throughout the penalization grid. This particular form of warm-starts is predominantly used in iterative algorithms for computing LS-EN estimates because it can drastically improve computation speed (e.g., Friedman et al. 2010). With the convex LS-EN objective function, the uncovered minima are actually global minima. In the context of robust estimators with non-convex objective function, the "0-based" regularization path is still usable but the uncovered minima, one per penalization level, are not necessarily

global minima. It is therefore necessary to also consider other initial estimates along the grid, such as initial estimates from random subsampling or the EN-PY procedure.

In Cohen Freue et al. (2019) we combine initial estimates from the EN-PY procedure with the idea of warm-starts. We take a small number, say $Q_{\text{I}} \ll Q$, of penalization levels from the large set $\mathscr{Q}$, denoted by $\mathscr{Q}_{\text{I}} \subset \mathscr{Q}$. Only at these few levels of penalization initial estimates are computed with the EN-PY procedure. When traversing the fine grid to compute local minima of the PENSE objective function, warm-starts at $\lambda_q \in \mathscr{Q}$ are combined with initial estimates from the EN-PY procedure if λ_q is also in $\mathscr{Q}_{\text{I}}$. Further increasing the probability of uncovering global minima, the grid $\mathscr{Q}$ is traversed in both directions. In the second pass in reverse direction, local minima at λ_q are used to initialize the PENSE estimate at λ_{q-1}. This combined strategy of bidirectional warm-starts and EN-PY effectively reduces computation while maintaining high quality of the uncovered minima.

Absent from the discussion so far, but critical for computing initial estimates, is the issue of translating a specific level of penalization of PENSE to comparable penalization of the initial estimates. Both procedures for computing initial estimates presented here use LS-EN estimates, computed on a subset of the data, to locate PENSE estimates nearby. For this to be successful, the amount of penalization induced by the penalty level λ_{I} on a LS-EN estimate compute on a (small) subset of the data, must approximately match the effect of the desired penalization level λ_{S} on the PENSE estimate computed on the full data. Because of the differences in loss function and data used for computation, using the same penalization level does not work well in general. Particularly the very different loss functions can lead to the empty model from the LS-EN estimate for any subset of the data for a certain λ, while a global optimum of the PENSE objective function at this λ corresponds to all predictors having non-zero coefficient estimate.

For the S-Ridge estimator, Maronna (2011) matches the regularization parameters between LS-Ridge and S-Ridge via a multiplicative adjustment of λ_{I} to get λ_{S}. The author derives these adjustment factors from the ratio of the squared M-scale estimate to the variance estimate of a Normal random variable in two extreme cases: (i) the mean of the Normal distribution is 0 and (ii) the variance of the Normal distribution is 0. For a given value of δ in the definition of the S-loss (2.8) these two ratios can be computed exactly, and the author takes the geometric mean of these two numbers as a crude approximation to the expected ratio of the S-loss to the LS-loss at their respective optima. The adjustment is easy to compute, but empirical observations suggest the quality of the match is suboptimal.

The combined strategy of warm-starts and EN-PY initial estimates in Cohen Freue et al. (2019) also suffers from an imperfect match of penalization levels. The effects, however, are less detrimental because local minima are followed across penalization levels. For computational reasons, however, not every local minimum is traced throughout the entire path, only the most promising minima. If the penalization introduced in the initial estimate is vastly different from the penalization of the PENSE estimate, this filter may drop minima prematurely. This problem can be avoided by merging all EN-PY initial estimates from each penalization level in $\mathscr{Q}_{\mathrm{I}}$ into one large set of initial estimates, $\mathscr{T}$. Each of these initial estimates is used for computing PENSE at every $\lambda_{\mathrm{S}} \in \mathscr{Q}$. Instead of relying on an approximate matching between λ_{S} for PENSE and the regularization parameter λ_{I} used for the initial estimate, the idea is that for each $\lambda_{\mathrm{S}} \in \mathscr{Q}$, there should be at least one $\lambda_{\mathrm{I}} \in \mathscr{Q}_I$ which gives roughly the same penalization of the initial estimate as λ_{S} provides for the PENSE estimate. Although the match will in general not be perfect, the chance that some of the initial estimates will be close to a global optimum are much higher if trying several different regularization parameters for the initial estimates. The chances can be increased even further by combining the set of initial estimates $\mathscr{T}$ with the idea of warm-starts. Empirically, this simplified scheme leads to slightly better optima than bidirectional warm-starts proposed in Cohen Freue et al. (2019). The computational burden of using this excessively large number of initial estimates can be contained by fully iterating only "promising" initial estimates. Because the simplified scheme is more amenable to algorithmic optimizations, computational complexity is very similar to bidirectional warm-starts. Further details about these optimizations to improve computational performance are given in Chapter 6.

3.3 Theoretical Properties

None of the discussed strategies for initial estimates can guarantee that a global optimum of the PENSE objective function is attained, but the chances are good if using EN-PY and, if enough computing resources are available, can be increased by adding initial estimates obtained from a large number of random subsamples. The global optimum, however, is desirable due to its provable statistical properties. In the following, the PENSE estimator $\tilde{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}^0 \in \mathbb{R}^{p+1}$ is defined as the global minimum of the PENSE objective function

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\mu,\boldsymbol{\beta}} \mathscr{O}_{\mathrm{S}}(\mu, \boldsymbol{\beta}; \lambda_{\mathrm{S},n}, \alpha_{\mathrm{S}}) \tag{3.4}$$

where α_S and $\lambda_{\mathrm{S},n}$ are independent of the given data, but $\lambda_{\mathrm{S},n}$ can depend on the number of observations n.

As detailed in the previous chapter, it is desired for the estimator to be consistent for the true regression parameters. To derive consistency of the PENSE estimator, several assumptions are imposed on the linear regression model (2.2):

[A1] $\mathbb{P}(\boldsymbol{\mathcal{X}}^\intercal\boldsymbol{\theta} = 0) < 1 - \delta$ for all non-zero $\boldsymbol{\theta} \in \mathbb{R}^p$ and δ as defined in (2.8).

[A2] The distribution F_0 of the residuals $\mathcal{U}$ has an even density $f_0(u)$ which is monotone decreasing in $|u|$ and strictly decreasing in a neighborhood of 0.

[A3] The second moment of G_0 is finite and $\mathbb{E}_{G_0}[\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^\intercal]$ is non-singular.

Assumption [A1] ensures that the probability that observations are perfectly aligned on a hyperplane is not too large. It is noteworthy that the assumption on the residuals [A2] does not impose any moment conditions on the distribution, which makes the following results applicable to extremely heavy tailed errors. Furthermore, unlike many results concerning regularized M-estimators, PENSE only requires finite second moment of the predictors.

The proofs of the following properties also require the ρ function to satisfy the condition that

[R4] $t\rho'(t)$, is unimodal in $|t|$. In other words, there exists a c' with $0 < c' < c$, where c is the threshold defined in [R2], such that $t\rho'(t)$ is strictly increasing for $0 < t < c'$ and strictly decreasing for $c' < t < c$.

Although this assumption is a slight variation of more common assumptions on the mapping $t \mapsto t\rho'(t)$, it is nevertheless satisfied by most bounded ρ functions used for robust estimation, including Tukey's bisquare function.

The results in Smucler and Yohai (2017) about the consistency of the S-Ridge can be applied directly to the PENSE estimator.

Proposition 1. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d observations with distribution G_0 which satisfies* (2.2)*. Under assumptions [A1] and [A2] and if $\lambda_{s,n} \to 0$, the PENSE estimator $\tilde{\boldsymbol{\theta}}$ as defined in* (3.4)*, is a strongly consistent estimator of the true regression parameter $\boldsymbol{\theta}^0$: $\tilde{\boldsymbol{\theta}} \xrightarrow{a.s.} \boldsymbol{\theta}^0$.*

Although the penalty functions used for the S-Ridge and PENSE are different, the growth condition on $\lambda_{\mathrm{S},n}$ has the same effect on PENSE as on the S-Ridge; making the

penalty term negligible for large enough n. The proof of Proposition (1) is therefore identical to the proof of Proposition 1.i in Smucler and Yohai (2017).

The next step is to quantify the speed of convergence in Proposition (1). The following theorem states that the PENSE estimate exhibits a $n^{1/2}$ converges to the true parameter.

Theorem 1. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d observations with distribution G_0 which satisfies* (2.2)*. Under regularity conditions [A1]–[A3] and if $\lambda_{s,n} = O(1/\sqrt{n})$, the PENSE estimator $\tilde{\boldsymbol{\theta}}$ as defined in* (3.4)*, is a root-n consistent estimator of the true parameter vector $\boldsymbol{\theta}^0$: $\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\| = O_p(1/\sqrt{n})$.*

The proof of this theorem is given in Appendix B.2.2 for a more general penalty function, of which the EN penalty is a special case. The proof is based on first-order Taylor expansions of the objective function around the true parameter $\boldsymbol{\theta}^0$ and the true residuals u_i.

Consistency and root-n consistency of PENSE both hold even under very heavy tailed error distributions F_0 and only require a finite second moment of the predictors. Importantly, the estimator is consistent for the true parameters without any prior knowledge about H_0; it is irrelevant whether the M-scale of the residuals is tuned to be a consistent estimator of the true scale of the error or not. Although the main focus of regularized estimators are applications with many predictors and a comparably small sample size, the asymptotic results in this section provide assurance that PENSE is sensible for estimating parameters in the linear regression model. Furthermore, the asymptotic guarantees for PENSE are necessary for developing theoretical results in the following chapter which allow for informative comparisons with other methods. The theory presented so far, however, does not specify how arbitrary contamination may affect the estimator.

3.4 Robustness

An overarching goal of this work is to devise estimators which can tolerate a considerable amount of contamination without giving aberrant results. Despite its shortcomings when it comes to regularized estimators as mentioned in Section 2.2, the finite-sample breakdown point is an important measure of robustness; it measures how much contamination can be introduced such that the maximum bias under contamination remains bounded.

An appealing property of the FBP is that it can usually be proven theoretically without resorting to numerical experiments. For PENSE, the breakdown point is close to δ as shown in the following theorem.

Theorem 2. *For a sample* $\mathscr{Z} = \{(y_i, \mathbf{x}_i) \colon i = 0, \ldots, n\}$ *of size* n*, let* $m(\delta) \in \mathbb{N}$ *be the largest integer strictly smaller than* $n \min(\delta, 1 - \delta)$*, where* δ *is as defined in* (2.8)*. Then, for a fixed* $\lambda_{S,n} > 0$ *and* $\alpha \in [0, 1]$*, the breakdown point* (2.7) *of the PENSE estimator* $\tilde{\boldsymbol{\theta}}$ *as defined in* (3.4)*,* $\epsilon^* \left(\tilde{\boldsymbol{\theta}}; \mathscr{Z} \right)$*, satisfies the following inequalities:*

$$\frac{m(\delta)}{n} \leq \epsilon^* \left(\tilde{\boldsymbol{\theta}}; \mathscr{Z} \right) \leq \delta \,.$$

The proof of this theorem can be found in the Appendix B.1.

The finite-sample breakdown point does not reveal the actual magnitude of the bias, MSE, or prediction error under contamination; it only states that these measures are finite for a contamination proportion less than δ. For applications, however, it is important to have a better understanding of an estimator's behavior under contamination. Numerical experiments, detailed in Section 3.6, shed light on the behavior of the PENSE estimator under contamination.

3.5 Hyper-Parameter Selection

The asymptotic properties of PENSE depend on an appropriate choice of the hyper-parameter $\lambda_{\mathrm{S},n}$. More specifically, the estimator is consistent only if $\lambda_{\mathrm{S},n}$ only if $\lambda_{\mathrm{S},n} \to 0$. In practice this growth rate is difficult to ascertain. Furthermore, while the theoretical properties do not depend on a certain choice of α, it nevertheless impacts the performance of the estimator. For the remainder of this section the subscripts of $\lambda_{\mathrm{S},n}$ are being dropped as only PENSE for a fixed sample size n is being considered.

3.5.1 Restricting the Search Space

Before discussion strategies for choosing the hyper-parameters, the search space needs to be restricted, in particular the range of values considered for the penalization level λ. For convex objective function, e.g., the LS-EN objective function, it is straightforward to determine the largest penalization level such that $\boldsymbol{\beta} = \mathbf{0}_p$ is a global minimum. It is unnecessary to consider penalization levels beyond this largest level, as the global minimum will be the same for all of them.

This upper bound cannot easily determined for the PENSE objective function due to the non-convexity of the problem. It is, however, possible to determine $\widetilde{\lambda}_{\mathrm{S}}$, the smallest penalization level such that $\boldsymbol{\beta} = \mathbf{0}_p$ is a local minimum, using the generalized gradient

as defined in Clarke (1990). First it is important to note that because the unpenalized S-loss is continuously differentiable and the EN penalty is locally Lipschitz, the PENSE objective function is also locally Lipschitz. Therefore, the generalized gradient of the PENSE objective function is the subgradient of the EN penalty plus the derivative of the S-loss. The subgradient of a convex function $g\colon \mathbb{R}^p \to \mathbb{R}$ at $\mathbf{u}_0$ is defined by Clarke (1990) as the set

$$\boldsymbol{\nabla}_{\mathbf{u}} g(\mathbf{u})\Big|_{\mathbf{u}=\mathbf{u}_0} = \{\mathbf{v}\colon \mathbf{v}^\intercal(\tilde{\mathbf{u}} - \mathbf{u}_0) \leq g(\tilde{\mathbf{u}}) - g(\mathbf{u}_0) \quad \forall \tilde{\mathbf{u}} \in \mathbb{R}^p\}\,.$$

Since the generalized gradient evaluated at any local minimum must contain $\mathbf{0}_{p+1}$, it is sufficient to determine the smallest penalty level such that the subgradient of the EN penalty, evaluated at $\boldsymbol{\beta} = \mathbf{0}_p$, contains the gradient of the S-loss evaluated at $\boldsymbol{\beta} = \mathbf{0}_p$, i.e.,

$$\widetilde{\lambda}_{\mathrm{S}} = \inf\left\{\lambda > 0\colon \nabla_{\boldsymbol{\beta}} \mathscr{O}_{\mathrm{S}}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta})\Big|_{\boldsymbol{\beta}=\mathbf{0}_p} \in \lambda \boldsymbol{\nabla}_{\boldsymbol{\beta}} \Phi_{\mathrm{EN}}(\boldsymbol{\beta}; \alpha)\Big|_{\boldsymbol{\beta}=\mathbf{0}_p}\right\}.$$

The subgradient of the EN penalty and the gradient of the S-loss are given by

$$\boldsymbol{\nabla}_{\boldsymbol{\beta}} \Phi_{\mathrm{EN}}(\boldsymbol{\beta}; \alpha)\Big|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = \left(\begin{cases} (1-\alpha)\tilde{\beta}_j + \alpha \operatorname{sgn}(\tilde{\beta}_j) & \tilde{\beta}_j \neq 0 \\ [-\alpha; \alpha] & \tilde{\beta}_j = 0 \end{cases}\right)_{j=1}^{p}$$

$$\nabla_{\boldsymbol{\beta}} \mathscr{O}_{\mathrm{S}}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta})\Big|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = -\frac{1}{n}\sum_{i=1}^{n} w_i^2(\mathbf{y} - \mu - \mathbf{X}\tilde{\boldsymbol{\beta}})\left(y_i - \mu - \mathbf{x}_i^\intercal \tilde{\boldsymbol{\beta}}\right)\mathbf{x}_i,$$

with weights $w_i(\mathbf{y} - \mu - \mathbf{X}\tilde{\boldsymbol{\beta}})$ as defined in (3.3). Evaluated at $\boldsymbol{\beta} = \mathbf{0}_p$, the subgradient of the EN penalty is the set $\{\mathbf{b}\colon |b_j| \leq \alpha, j = 1, \ldots, p\}$. Combined with the gradient of the S-loss at $\boldsymbol{\beta} = \mathbf{0}_p$, $\widetilde{\lambda}_{\mathrm{S}}$ is therefore

$$\widetilde{\lambda}_{\mathrm{S}} = \frac{1}{n\alpha} \max_{j=1,\ldots,p} \left|\sum_{i=1}^{n} w_i^2(\mathbf{y} - \hat{\mu}_y)(y_i - \hat{\mu}_y)x_{ij}\right|, \tag{3.5}$$

where $\hat{\mu}_y$ is the estimated intercept in the empty model, $\hat{\mu}_y = \arg\min_\mu \hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \mu)$. If $\widetilde{\lambda}_{\mathrm{S}} > 0$, the 0-vector is a local minimum of $\mathscr{O}_{\mathrm{S}}(\mu, \boldsymbol{\beta}; \lambda, \alpha)$ for all $\lambda > \widetilde{\lambda}_{\mathrm{S}}$. On the other hand, if $\widetilde{\lambda}_{\mathrm{S}} = 0$, the 0-vector is a local maximum for all λ smaller than a certain value L and a local minimum for $\lambda > L$. In this border case, no simple expression exists to determine L and a trial and error search for $\widetilde{\lambda}_{\mathrm{S}}$ is the only other option.

With the approximate upper bound $\widetilde{\lambda}_{\mathrm{S}}$, the search for an optimal penalization level can be concentrated on the range $(0, \widetilde{\lambda}_{\mathrm{S}})$. The prevalent strategy is to tune the hyper-parameters

to optimize some performance metric of interest, such as metrics pertaining to the quality of the fit or the prediction performance. Robust fit-based metrics, for example robust versions of popular information criteria AIC (Akaike 1974) or BIC (Schwarz 1978), rely on a robust estimate of the residual scale. In high-dimensional settings, however, estimating the residual scale is a difficult task by itself. Especially robust estimation is challenging, because robust scale estimates themselves require tuning parameters. Changing these tuning parameters is effectively changing the information criterion itself; if the distribution of the error term is unknown, there is no general way to choose these tuning parameters.

More importantly, applications motivating this work demand estimators with strong prediction performance. For these applications, fit-based metrics are not useful because they only give limited insight into how well the fitted model generalizes beyond the sample at hand. Due to this shortcoming, prediction performance is usually evaluated using measures of the prediction error. The prediction error, however, cannot be sensibly estimated on the same data as used to fit the model, i.e., the data used in the computation of PENSE. Strategies to estimate the prediction error are most often based on withholding some of the available observations (i.e., the "test" set) and computing optima of the PENSE objective function on the remaining observations (i.e., the "training" set). The prediction error is then estimated as the error arising by predicting the responses of the withheld observations.

3.5.2 Cross Validation

The arguably most prevalent strategy for estimating prediction performance is K-fold cross validation (CV). In K-fold CV, the n observations in the sample at hand are split into K disjoint sets of roughly equal size, called folds. In cross-validation, every observation is used exactly once for prediction and $K-1$ times for training, i.e., computing of a global optimum of the objective function.

To outline the procedure, the index set of a single fold is denoted by $\mathscr{S}_k \subset \{1,\dots,n\}$, $k = 1,\dots,K$. These sets are such that they are disjoint, roughly the same size, and $\bigcup_{k=1}^{K} \mathscr{S}_k = \{1,\dots,n\}$. For each $k \in \{1,\dots,K\}$, a global optimum of the objective function using the observations in $\bigcup_{k'=1,k'\neq k}^{K} \mathscr{S}_{k'}$ is computed and denoted by $\hat{\boldsymbol{\theta}}_k^{(\lambda,\alpha)}$. These k optima are used to predict the observed responses in the k-th fold by

$$\hat{y}_i^{(\lambda,\alpha)} = \mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}_k^{(\lambda,\alpha)} + \hat{\mu}_k^{(\lambda,\alpha)} \quad \text{for all } i \in \mathscr{S}_k. \tag{3.6}$$

No observation affects the optimum used to predict its value and hence these n predicted

values can be used to adequately estimate the prediction error of the method with hyper-parameters (λ, α).

A popular metric for the prediction performance an estimator $\hat{\boldsymbol{\theta}}$ is its root mean squared prediction error (RMSPE), defined as

$$\text{RMSPE}\left(\hat{\boldsymbol{\theta}}\right) = \sqrt{\mathbb{E}\left[(\mathcal{Y} - \boldsymbol{\mathcal{X}}^\intercal \hat{\boldsymbol{\beta}} - \hat{\mu})^2\right]}. \tag{3.7}$$

Using cross-validation, the RMSPE can be estimated by

$$\widehat{\text{RMSPE}}(\lambda, \alpha) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i^{(\lambda,\alpha)} - y_i\right)^2}.$$

If the error distribution is heavy-tailed the RMSPE might not be well defined. More importantly, under the presence of contamination in the sample the estimated RMSPE is badly affected and does not adequately reflect the estimate's prediction performance. Since the RMSPE is essentially a measure of the expected absolute size of the prediction error, it is more sensible to use a robust measure of scale for quantifying the prediction performance. A common choice to robustly measure the prediction performance of an estimator $\hat{\boldsymbol{\theta}}$ is the uncentered τ-scale (Maronna and Zamar 2002) of the prediction errors, given by

$$\tau_{\text{P}}\left(\hat{\boldsymbol{\theta}}\right) = \sqrt{\mathbb{E}\left[\max\left(c_\tau, \frac{\left|\mathcal{Y} - \boldsymbol{\mathcal{X}}^\intercal \hat{\boldsymbol{\beta}} - \hat{\mu}\right|}{\text{Median}\left|\mathcal{Y} - \boldsymbol{\mathcal{X}}^\intercal \hat{\boldsymbol{\beta}} - \hat{\mu}\right|}\right)^2\right]}, \tag{3.8}$$

which can be estimated via CV by

$$\widehat{\tau}_{\text{P}}(\lambda, \alpha) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\max\left(c_\tau, \frac{\left|y_i - \hat{y}_i^{(\lambda,\alpha)}\right|}{\underset{i'=1,\ldots,n}{\text{Median}}\left|y_i - \hat{y}_i^{(\lambda,\alpha)}\right|}\right)^2}.$$

The parameter $c_\tau > 0$ controls the tradeoff between efficiency and robustness of the τ-size by defining what constitutes outlying values in terms of multiples of the median absolute deviation. In this work, the τ-size is always reported for $c_\tau = 3$.

Once the set of hyper-parameters resulting in the best prediction performance is determined, a global optimum at these chosen hyper-parameters is computed using all n observations. Cross-validation is shown to work very well for regularized estimators using

convex objective functions (Hirose et al. 2013; Homrighausen and McDonald 2016; Homrighausen and McDonald 2018). Cross-validation performs well when a global optimum computed using all n observations, $\hat{\boldsymbol{\theta}}^{(\lambda,\alpha)}$, is "reasonably close" to global optima computed on the subsets of observations, $\hat{\boldsymbol{\theta}}_k^{(\lambda,\alpha)}$, $k = 1, \ldots, K$. This is usually the case if the amount of penalization induced by the hyper-parameters (λ, α) is comparable between the subsamples and the objective function only exhibits a single optimum. With non-convex objective functions, however, it is possible that a local optimum of the objective function evaluated on the full data is a global optimum when evaluated on a subset of the data. An example of this behavior is given in Figure 3.3 for data generated by a simple linear regression model with true parameter value $\beta^0 = 1$ and 30% of the observations contaminated. While the global minimum of the PENSE objective function evaluated on all observations is around 0.9, the global minimum of the objective function evaluated on three of the five subsets is close to -1. The subsets in this example satisfy the conditions for cross-validation and contamination never exceeds the desired breakdown point of 50%, but it is obvious that the predictions from three of the five estimates are likely far off. For this particular set of hyper-parameters the estimated prediction performance is therefore not representative of the prediction performance of the global minimum on the full data. Although this example shows an extreme scenario, it highlights that cross-validation may give very different estimates of the prediction performance for different splits of the data. This issue is not unique to PENSE, but any estimator defined via non-convex objective functions because of the disconnect between the minima uncovered in the CV folds and the estimate from the full data.

3.5.3 Train/Test Split

The challenges of cross-validation exposed in the previous section can be traced back to two issues: (i) estimating the prediction performance by combining the prediction errors from different optima (computed on different subsets of the data) which may not be comparable and (ii) trusting that this estimated prediction performance is representative of the prediction performance of the optimum computed on the full data set for the selected hyper-parameters.

These challenges could be surmounted by gauging the prediction performance of every possible estimate directly. For train/test splitting, PENSE estimates are computed on a random subset of the data (i.e., the training set) and the estimates' prediction performance is evaluated on the left-out observations (i.e., the test set). In contrast to cross-validation,

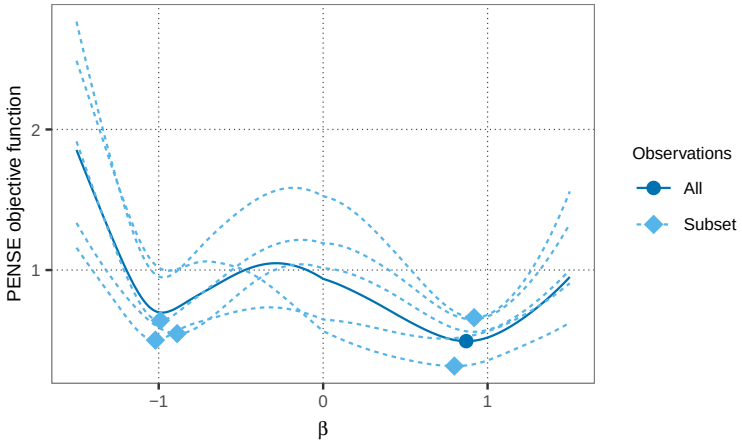


Figure 3.3: PENSE objective function (3.1) for a simple linear regression model of the form $\mathbf{y} = \mathbf{x} + \mathbf{u}$, evaluated at different values of β on the full data set with 100 observations (solid blue line) and subsets of size 80 (dashed light blue lines). The points on each curve mark the global minimum of the objective function evaluated on the particular subset.

the PENSE estimates are not computed on the full data set but only on the training set, avoiding the issues highlighted before.

Simple train/test splitting, however, suffers from different issues, especially in the presence of contamination. If there is a large number of contaminated observations in the test set, it is not possible to accurately estimate the prediction performance of the PENSE estimates. Estimates which are affected by contamination in the training set may appear to have good prediction performance. On the other hand, "good" PENSE estimates will not appear as such since contaminated observations in the test set will not be predicted well. A single train/test split is therefore not sufficient.

It is more appropriate to equally divide the observations into K disjoint folds, similar to cross-validation. Each fold is used as test set exactly once, with the remaining $K - 1$ folds being used for training. This leads to K PENSE estimates for every hyper-parameter-configuration, with each estimate being evaluated on a different test set. If the total contamination in the data is ϵn, there is at least one test set with less than $\epsilon n K/(K-1)$ contaminated observations. Nevertheless, an estimate affected by contamination can still appear to outperform the other estimates.

A more resilient procedure can be constructed when averaging comparable information from all K folds. As outlined above PENSE estimates computed with the same λ, but on different subsets of the data, might not be comparable. The effect of α, on the other hand, is more stable across subsets of the data. The following two-stage procedure therefore leads

to a more stable hyper-parameter selection than simple train/test splitting. For each α in a grid of values, $\mathscr{A} = \{\alpha_1, \ldots, \alpha_A\}$, and for every fold $k = 1, \ldots, K$, select the PENSE estimate with hyper-parameter λ_k which minimizes the scale of the prediction error in the k-th fold. Thus, each of the K folds yields A PENSE estimates, one for every $\alpha \in \mathscr{A}$. Test sets with a large proportion of contamination can occasionally lead to a highly underestimated scale of the prediction error. If the breakdown point of the estimator, however, is large enough, it is unlikely that this phenomenon occurs for every α in the grid. The prediction performance in each of the K folds can be summarized by taking the median scale of the prediction error of all A estimates in the k-th fold. The final PENSE estimate is then chosen as the estimate with minimum scale of the prediction error in the fold with smallest median scale of the prediction error.

The major drawback of train/test splitting is that some observations are forfeited for use as test set. While this can improve estimation of the prediction performance, it can directly lower the prediction performance of the PENSE estimate, because it does not have access to all observations. The numerical experiments conducted in the following section also expose this weakness of train/test splitting. Although CV is sometimes much more affected by contamination, in the majority of cases estimates computed by train/test splitting seem to be slightly worse.

3.6 Numerical Experiments

The theoretical properties in Section 3.3 give an indication about the qualities of the PENSE estimator, but it is difficult to translate these asymptotic properties into tangible metrics on finite samples. The growth condition on the penalty parameter $\lambda_{\text{s},n}$, for example, requires a procedure independent of the data to select the penalty parameter; there are no theoretical guarantees regarding the data-driven hyper-parameter selection procedures outlined in Section 3.5. Similarly, the breakdown point of PENSE only guarantees that the parameter estimates remain bounded, but it is unknown how contamination affects the estimates. Numerical experiments are a useful tool to gauge the effectiveness of different hyper-parameter selection strategies and the practical performance and robustness of PENSE and competing estimators.

3.6.1 Estimators

In the following experiments, PENSE is computed with a breakdown point of 33%, i.e., δ in the S-loss (2.9) is set to 0.33. The grid of α values is $\mathscr{A} = \{0.5, 0.66, 0.83, 1\}$ and the grid for λ comprises 50 values equidistant on the log-scale with the upper endpoint $\widetilde{\lambda}_{\text{S}}$ (derived in Section 3.5.1) and the lower endpoint set to $0.001\alpha\widetilde{\lambda}_{\text{S}}$. Initial estimates for PENSE are computed according to the 0-based regularization path and the simplified scheme described in Section 3.2.4, for a total of 10 penalization levels. As justified by the results in the beginning of Section 3.6.3, the hyper-parameters α and λ are selected by 5-fold cross-validation as discussed in Section 3.5. Prediction performance is measured by the τ-scale of the prediction errors. A detailed description of the algorithms used to compute the PENSE estimate is given in Chapter 6.

PENSE is compared to several other robust and non-robust estimators. The most similar robust estimator to PENSE is MMLASSO, with the initial S-Ridge estimate computed for 10 different penalization levels and the penalization level for MMLASSO selected by 5-fold CV. In low- to moderate-dimensional settings only ($p < (1 - \delta)n - 1$), the robust unregularized S- and MM-estimators (denoted by *S* and *MM*, respectively) are computed as provided in the R-package `RobStatTM` (Yohai et al. 2019), with breakdown point set to 33%. For hypothetical comparisons, the oracle S- and MM-estimates are computed using only the truly active predictors. All robust estimates employ Tukey's bisquare ρ function, with cutoff set to 2.37 which yields a consistent scale estimate in case of Normal errors and $\delta = 0.33$.

The LS-EN estimate is computed using the `glmnet` (Simon et al. 2011) R package. Hyper-parameters are selected by 5-fold CV on the same grid of α values as used for PENSE and the penalty parameter λ_{LS} is chosen from a set of 50 values generated by `glmnet`. Prediction performance for cross-validation is measured by the mean absolute prediction error.

3.6.2 Scenarios

Robust estimators should perform well under any conceivable contamination. While it is infeasible to cover every possible contamination, the objective function of PENSE suggests the kind of contamination with most severe effect on the estimate. As for other S-estimators of linear regression (e.g., Maronna 2011), a strong linear relationship between the contaminated responses and predictors combined with high leverage potentially leads to a large bias

in the PENSE estimate. The numerical experiments in this section therefore cover a range of contamination scenarios where the contaminated observations follow a linear relationship different from the majority of the data.

The majority of the n observations follows the linear model

$$y_i = x_{i1} + \cdots + x_{is} + u_i \qquad i = \lfloor \epsilon n \rfloor + 1, \ldots, n$$

where $\mathbf{x}_i$ is the vector of p predictors following a multivariate t-distribution with 4 degrees of freedom and $s < p$ is the number of predictors with non-zero coefficient. The error terms u_i are i.i.d. following a stable distribution with varying tail parameter. The empirical scale of the error term, $\hat{\sigma}_{\mathbf{u}}$, is chosen to control for the proportion of variance explained by the model (PVE), ν:

$$\nu = 1 - \frac{\hat{\sigma}^2_{\mathbf{u}}}{\hat{\sigma}^2_{\mathbf{y}}}.$$

Following the argument in Hastie et al. (2017) on realistic values of explained variation, ν is fixed at 0.25.

Contaminated observations, on the other hand, follow the linear model

$$y_i = k_{\text{v}} \tilde{\mathbf{x}}_i^\intercal \boldsymbol{\pi} + u'_i \qquad i = 1, \ldots, \lfloor \epsilon n \rfloor$$

with parameter k_{v} controlling the "outlyingness" of the contaminated observations and perturbation u'_i following a centered Normal distribution scaled such that the model explains 91% of the variation in the contaminated observations. On the one hand, large values of $|k_{\text{v}}|$ lead to farther outlying observations and hence have more potential of biasing estimates. On the other hand, robust estimators can better identify highly outlying observations as contaminated and assign low weights in the estimation to these observations. Additionally, the regularizing term steers the estimate towards the model favoring the non-contaminated observations if $|k_{\text{v}}|$ is very large. Therefore, it is difficult to predict which values of k_{v} lead to higher bias of PENSE estimates. To get an overall assessment of the bias incurred by contamination, five different contamination parameters k_{v} are considered, $k_{\text{v}} = \{-2, -1, 0, 3, 7\}$.

The vector $\boldsymbol{\pi}$ is randomly generated to have exactly s 1's and $p - s$ 0's, determining which predictors are included in the linear relationship of the contaminated observations. Leverage of the contaminated observations is increased by scaling the values of the predictors included in the linear model for the contaminated observations. The magnitude of scaling is

determined by contamination parameter $k_l > 1$. Larger values of the scaling factor k_l lead to higher leverage of the contaminated observations and thus to larger bias of estimates, but the effect on robust estimates levels off. Therefore the value is fixed for all contamination scenarios at $k_l = 8$. The detailed scaling mechanism is explained in Appendix A.3.

Scenarios without contamination ("no contamination") are replicated 100 times, while contaminated scenarios are replicated 50 times. A detailed description of the simulation scenarios and data generation schemes is given in Appendix A.3.

3.6.3 Results

Before comparing PENSE to other regularized estimators, the strategy for hyper-parameter selection is to be determined. Figure 3.4 shows the relative scale of the prediction error for PENSE estimates obtain from different hyper-parameter selection strategies in all considered scenarios. To compare results across error distributions, sample sizes, and sparsity settings, the scale of the prediction error is standardized by the scale of the prediction error of the PENSE estimate obtained from hyper-parameters selected to minimize the prediction error on a large independent validation set. This validation set is in practice unavailable but is the gold-standard to compare the different hyper-parameter selection strategies.

The strategies compared in Figure 3.4 are cross-validation (Section 3.5.2) and train/test splitting as outlined in Section 3.5.3. The strategy *Train/Test (min)* uses the estimate resulting in the smallest estimated prediction error, while *Train/Test (avg)* averages information from all K folds, as detailed at the end of Section 3.5.3. Figure 3.4 highlights that CV is preferable to train/test splitting in the vast majority of cases. Especially for scenarios without contamination, the PENSE objective function is in general well-behaved and local minima do not cause problems, while train/test splitting clearly suffers from a reduced sample size. Under contamination, CV still tends to perform better than train/test splitting, albeit the difference is in general negligible. The numerical experiment also underlines that, in isolated cases, CV does suffer from the issues outlined in Section 3.5.2. Overall, however, the benefits of cross-validation and using the full sample to compute the estimates dominate train/test splitting strategies.

Summarizing results under contamination

Scenarios with 25% contamination may be grouped into groups of five scenarios by ignoring the value of the contamination parameter, k_v. In each of these 5 scenarios, the uncontam-

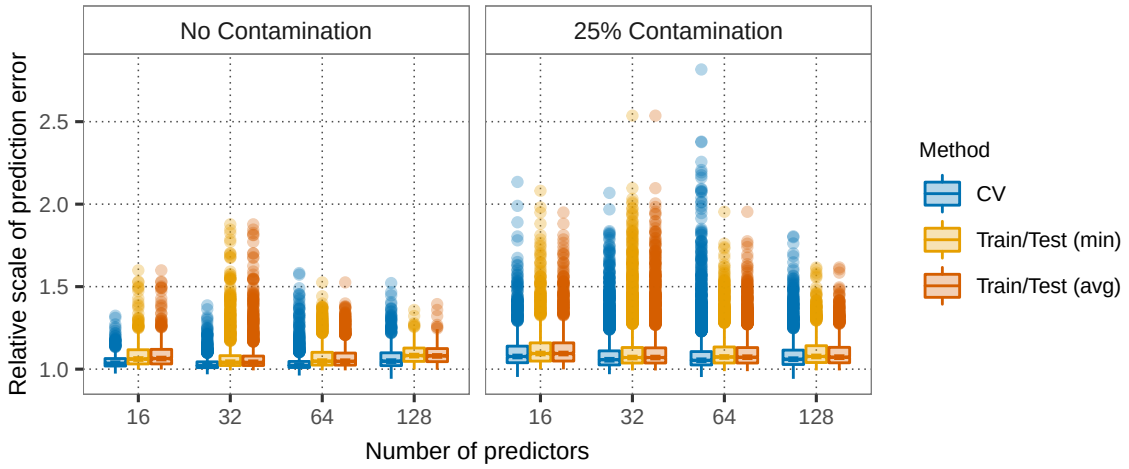


Figure 3.4: Prediction performance of PENSE estimates with hyper-parameters chosen according to 5-fold CV or different versions of 5-fold train/test split. The scale of the prediction error on the vertical axis is shown relative to the prediction error of the PENSE estimate with hyper-parameters obtained by using an independent validation set of 1000 observations. The boxplots include results from all considered scenarios.

inated observations are identical and the same as in the corresponding scenario without contamination. Figure 3.5 shows the τ-size of the prediction error estimated on an independent validation set relative to the true scale of the residuals for LS-EN, MMLASSO, and PENSE, under the different outlier positions. In this plot, an outlier position of $k_v = 1$ corresponds to the scenario without contamination. For the non-robust LS-EN estimator, prediction performance decreases sharply with increasing severity of the outliers, i.e., $|k_v - 1|$, but the effects are much more pronounced in the very sparse scenario *VS1-MH(k_v, 8)* shown in the left panel. The robust estimators MMLASSO and PENSE show similar performance across different outlier positions, but MMLASSO seems to be more affected by some of the outlier positions than PENSE. Both robust estimators are most affected by moderate severity of the outliers, i.e., contamination which is not easily detectable as such, but neither exhibits a severe loss of prediction performance.

Contamination in sparse scenarios (i.e., 24 predictors out of 64 are active) appears to be less problematic than in very sparse scenarios (6 predictors are active). The reason for this phenomenon is that in sparse scenarios the true scale of the residuals is much greater than in very sparse scenarios as the proportion of variance explained is kept constant at $\nu = 0.25$. The increased true error scale results in the residuals of the contaminated observations (with regards to the true model) being much less extreme as for similar very sparse scenarios. Therefore, neither robust nor non-robust estimators are highly affected by

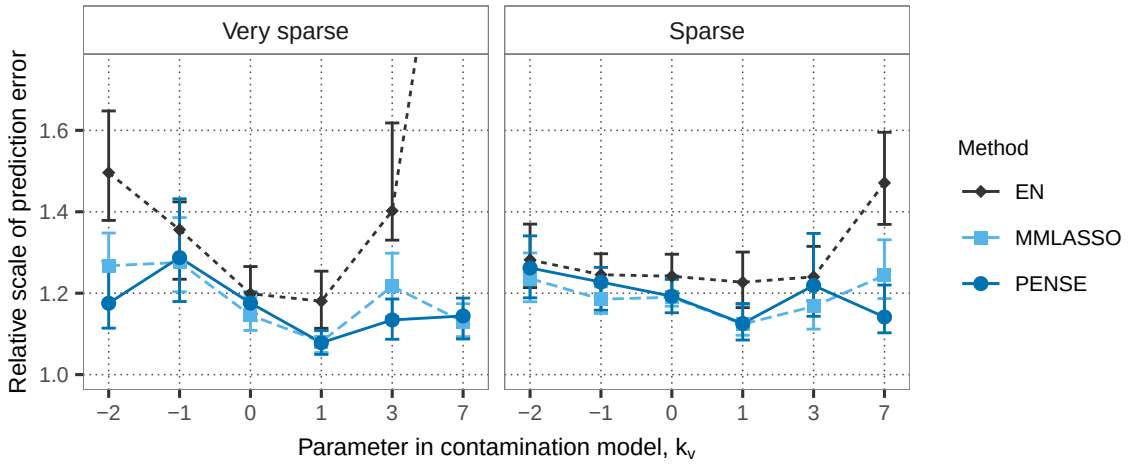


Figure 3.5: Prediction performance of regularized estimators under scenarios *VS1-MH(k_v, 8)* (left) and *MS1-MH(k_v, 8)* (right) with $n = 100$ and $p = 64$. The horizontal axis shows the different outlier positions, k_v, where $k_v = 1$ corresponds to the "no contamination" scenario. The scale of the prediction error on the vertical axis is shown relative to true scale of the residuals. The error bars depict the range of the inner 50% (inner quartile range) of relative prediction errors from 50 replications.

the contamination in sparse scenarios. The trend of the LS-EN estimator, however, strongly indicates that for larger values of k_v the estimator will lead to nonsensical predictions.

For an overall assessment of performance of the estimators under contamination, the metrics reported below are summarizing the scenarios by ignoring the value of the contamination parameter k_v. In other words, the different outlier positions are treated equally when assessing performance. Scenarios with contamination are replicated 50 times, and hence the reported values summarize $5 \times 50 = 250$ values. This leads to simpler comparison of different methods across different scenarios based on their "average performance under contamination".

Prediction performance

Prediction performance is measured either by the root mean square prediction error (RMSPE) as defined in (3.7) or by the τ-size of the prediction errors, defined in (3.8). The RMSPE is standardized by the empirical standard deviation of the true errors $\hat{\sigma}_{\mathbf{u}}$ and reported only for Normal errors. The τ-size, standardized by the empirical τ-scale of the true errors, $\hat{\tau}_{\mathbf{u}}$, is reported for all other error distributions without finite variance. Both measures of prediction performance are estimated on an independent test set of 1000 observations without contamination. A relative scale of the prediction error of 1 says the prediction error is of the same magnitude as the random error and indicates good prediction performance,

while larger values mean worse prediction performance.

Figure 3.6 shows boxplots of prediction performance for the LS-EN estimator, MM-LASSO, and PENSE under Normal and Cauchy errors and increasing number of predictors p. In all of these scenarios, the number of observations is fixed at $n = 100$ and the true model explains 25% of the variance in the observed response.

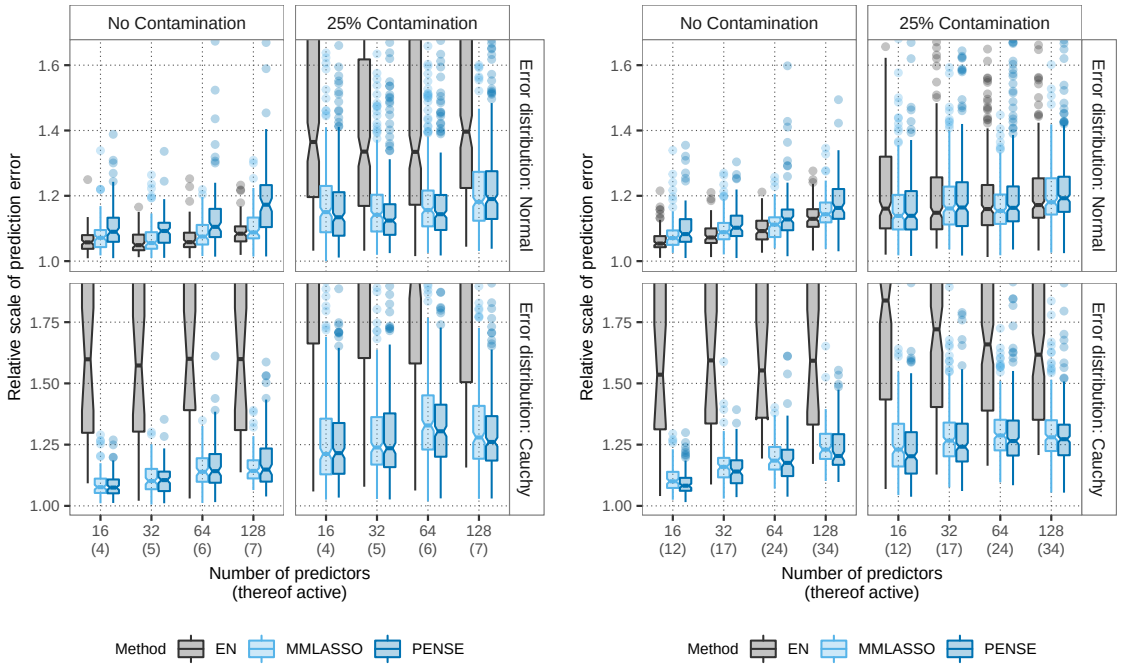
With more predictors available, the problem becomes more challenging and the prediction performance decreases accordingly. Even for low-dimensional problems, the prediction performance of the non-robust LS-EN estimator deteriorates drastically under the presence of contamination or heavy-tailed errors. Of the two robust estimators shown, MMLASSO leads to better prediction performance than PENSE for Normal errors and without contamination present, regardless of the number of truly active predictors. This can be expected since the scale estimate used for MMLASSO is tuned for consistency under Normal errors, leading to improved efficiency. For more heavy-tailed errors, however, the advantage of the M-step dissipates and the prediction performance of PENSE estimates is as good as or slightly better than MMLASSO estimates. While PENSE is outperformed by MMLASSO in some scenarios with Normal errors, MMLASSO seems more affected than PENSE by heavy-tailed errors and under the presence of grossly contaminated observations.

A comprehensive summary of the prediction performance in all scenarios, including additional error distributions and sample sizes, is given in Appendix C.1.1. It should be noted that in these visualizations LS-EN seems only slightly affected by contamination in sparse settings. As explained above, this is an artifact of the contamination being overshadowed by the large variability of the error term. Overall PENSE is the most stable of the considered estimator, leading to more robust estimates with highly competitive prediction performance.

Variable selection performance

The stated goal of PENSE is to achieve good prediction performance while at the same time identify relevant variables. For variable selection two measures are of interest: the relative number of correctly identified active predictors (sensitivity, SE) and the relative number of correctly identified inactive predictors (specificity, SP). These two measures are defined as

$$\mathrm{SE}(\hat{\boldsymbol{\beta}}) = \frac{\mathrm{TP}(\hat{\boldsymbol{\beta}})}{\mathrm{TP}(\hat{\boldsymbol{\beta}}) + \mathrm{FN}(\hat{\boldsymbol{\beta}})}\,, \qquad \mathrm{SP}(\hat{\boldsymbol{\beta}}) = \frac{\mathrm{TN}(\hat{\boldsymbol{\beta}})}{\mathrm{TN}(\hat{\boldsymbol{\beta}}) + \mathrm{FP}(\hat{\boldsymbol{\beta}})} \tag{3.9}$$



(a) Very sparse scenarios *VS1-LT** and *VS1-HT**.

(b) Sparse scenarios *MS1-LT** and *MS1-HT**.

Figure 3.6: Prediction performance of regression estimates in different scenarios with a sample size of $n = 100$. The horizontal axis in each panel shows the total number of predictors, while the vertical axis in each panel shows the root mean square prediction error (for Normal errors) or the τ scale of the prediction errors (for Cauchy errors).

where

$$\mathrm{TP}(\hat{\boldsymbol{\beta}}) = \left|\{j\colon \hat{\beta}_j \neq 0 \wedge \beta_j^0 \neq 0\}\right| \qquad \mathrm{FP}(\hat{\boldsymbol{\beta}}) = \left|\{j\colon \hat{\beta}_j \neq 0 \wedge \beta_j^0 = 0\}\right|$$
$$\mathrm{TN}(\hat{\boldsymbol{\beta}}) = \left|\{j\colon \hat{\beta}_j = 0 \wedge \beta_j^0 = 0\}\right| \qquad \mathrm{FN}(\hat{\boldsymbol{\beta}}) = \left|\{j\colon \hat{\beta}_j = 0 \wedge \beta_j^0 \neq 0\}\right|$$

are the number of true positives, false positives, true negatives, and false negatives, respectively. Perfect variable selection is achieved if both measures are 1, i.e., all active predictors have non-zero coefficient and all inactive coefficients have a coefficient value of 0.

Figure 3.7 shows the sensitivity and specificity under very sparse and sparse scenarios for a sample size of $n = 100$. As for prediction performance, variable selection is more challenging when more predictors are available and the more predictors are truly active. Variable selection of the non-robust LS-EN estimator is much more affected by heavy-tailed error distributions than by gross contamination. Particularly sensitivity drops to almost 0% for the LS-EN estimator if the errors are Cauchy distributed, even under no contamination;

the LS-EN estimate almost always selects the empty model in these scenarios. Interestingly, contamination by leverage points appears to help LS-EN identify some relevant predictors even for Cauchy errors. The reason is that some of the truly active predictors are contaminated by high-leverage values, which are immediately selected by LS-EN, alongside the other contaminated predictors. This highlights the hypersensitivity of LS-EN estimates to leverage point contamination; any predictor with leverage points will be selected by LS-EN with near certainty.

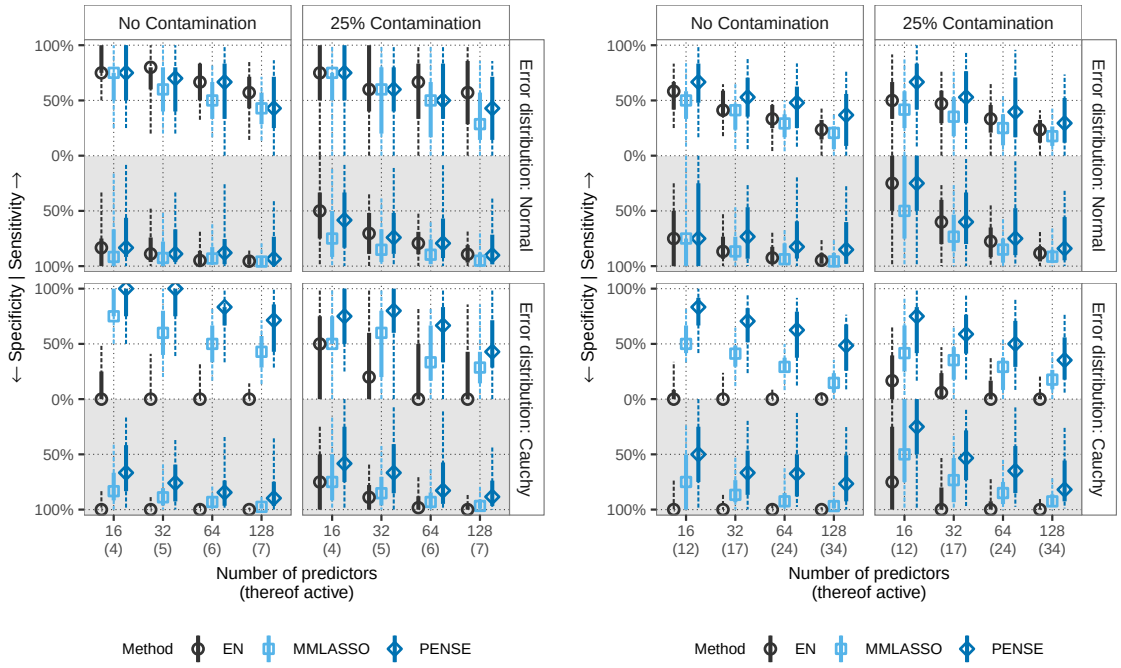
The robust estimators, on the other hand, perform very similarly for light- and heavy-tailed errors as well as under contamination. Sensitivity of both PENSE and MMLASSO estimates is almost unaffected by gross contamination under Normal errors and decreases only slightly if errors are Cauchy distributed. Specificity decreases more under contamination and it seems even robust estimators tend to wrongly select inactive predictors if they are contaminated with leverage points.

In sparse scenarios (right plot 3.7(b)), variable selection is apparently more challenging than in very sparse scenarios. The greater flexibility of the EN penalty used by PENSE seems to be an advantage in these sparse scenarios. While MMLASSO has comparable sensitivity to PENSE in very sparse scenarios, the L_1 penalty can be too restrictive for scenarios where many predictors are truly active. In these scenarios PENSE has substantially higher sensitivity than MMLASSO.

Across all scenarios, PENSE has the highest sensitivity and selects more of the truly active predictors than the other estimators. Even under no contamination and Normal errors, PENSE is as good as LS-EN in detecting truly active predictors. Unsurprisingly, MMLASSO tends to have lower sensitivity than PENSE because of the restrictions imposed by the L_1 penalty. On the other hand, PENSE usually selects many more irrelevant variables than MMLASSO. Overall, PENSE has high sensitivity but only moderate specificity, a shortcoming addressed in the following Chapter 4.

These conclusions also extend to the other error distributions and sample sizes, as visualized in Appendix C.1.2. Variability of the variable selection performance of PENSE estimates decreases substantially with larger sample size, and sensitivity improves noticeable. Specificity, on the other hand, increases only moderately with a larger sample size. Importantly, variable selection properties of LS-EN deteriorate quickly even for a moderate-light-tailed error distribution.

It is important to note that none of the estimators shown here possess any theoretical guarantees of uncovering the true active set with high probability in the scenarios considered.



(a) Very sparse scenarios *VS1-LT** and *VS1-HT**.

(b) Sparse scenarios *MS1-LT** and *MS1-HT**.

Figure 3.7: Variable selection performance of regularized regression estimates in different scenarios with a sample size of $n = 100$. The horizontal axis in each panel shows the total number of predictors. The vertical axis in each panel is split in two halves: sensitivity (i.e., the number of correctly identified active predictors) is shown on the top half, and specificity (i.e., the number of correctly identified inactive predictors) is shown downwards with perfect specificity (100%) on the bottom. Solid vertical lines show the range of the inner 50%, while the dashed lines extend from the 5% to the 95% quantile.

Regularized robust estimators with better variable selection performance are discussed in the following Chapter 4.

3.7 Conclusions

The elastic net S-estimator, PENSE, proposed in Cohen Freue et al. (2019) and explained in detail in this chapter, is a highly robust method for linear regression problems with favorable prediction performance and good variable selection properties. Compared to competing methods, PENSE does not require an auxiliary scale estimate and theoretical guarantees do not depend on moment conditions on the error term. This make PENSE a very versatile method applicable to problems with high noise and possible contamination in the response and the predictors.

PENSE gains its robustness towards contamination and heavy-tailed error distribu-

tions by regularizing the robust, non-convex S-loss with the EN penalty. Locating a good minimum of this non-convex objective function with limited computing resources requires carefully chosen initial estimates. Using ideas from the Peña-Yohai estimator (Peña and Yohai 1999), we devised the EN-PY procedure in Cohen Freue et al. (2019) for PENSE to compute initial regularized estimates based on subsets of the data which likely exclude observations with high leverage. In practice, the EN-PY procedure, outlined in Section 3.2.2, often leads to better local optima than other strategies to obtain initial estimates while being computationally much more efficient.

Despite the complications introduced by the non-convex objective function, I establish the root-n consistency of the PENSE estimator in Section 3.3 for a fixed number of predictors but otherwise very mild assumptions. These asymptotic results, however, require penalty parameters chosen independently of the available sample according to the necessary growth conditions. In practice, this is infeasible. Section 3.5 therefore discusses different data-driven strategies to select hyper-parameters based on the prediction performance of the resulting estimate. All of these heuristics are prone to high variability in the estimated prediction performance due to the potential presence of contaminated observations combined with the non-convex objective function. The numerical experiments suggest that hyper-parameters selected via cross-validation lead to better estimates than hyper-parameters selected by other data-driven methods in the vast majority of cases. While in rare cases CV seems to be more affected by contamination than train/test splitting, the overall performance of CV justifies its use in practice. This underlines that hyper-parameter selection is challenging for estimators defined through non-convex objective functions and becomes more challenging the more severe the non-convexity caused by contaminated observations.

The numerical experiments also demonstrate that PENSE leads to better prediction performance than other estimators with provable theoretical guarantees in problems with high noise in the response and/or contaminated observations. Besides the numerical experiments conducted and explained herein, empirical results in Cohen Freue et al. (2019) for different data generation schemes underscore the versatility of PENSE, especially in problems where some predictors are highly correlated. From these empirical results and from the theoretical results presented and developed in this chapter, it can be concluded that PENSE has strong prediction performance and estimation accuracy even under very challenging circumstances. None of the competing methods is able to cope with high noise levels and contamination both in the response and the predictors as good as PENSE.

With respect to variable selection, the simulation study shows that PENSE has very

high sensitivity in almost all scenarios. This high sensitivity, however, comes at the price of a large number of falsely selected predictors. In many applications, a large number of false positives is undesirable. In biomarker discovery studies, for instance, too many potential biomarkers lead to prohibitively expensive follow-up validation studies or render the biomarkers infeasible for clinical use. It is therefore of practical importance to develop a robust estimator with better variable selection performance, particularly higher specificity, without sacrificing sensitivity or prediction performance.

Chapter 4

Variable Selection Consistent S-Estimators

The penalized elastic-net S-estimator (PENSE), as detailed in the previous chapter, achieves highly robust estimation and prediction performance. Theoretical results and numerical experiments demonstrate that PENSE estimates yield competitive prediction performance outperforming other estimators in challenging problems with heavy-tailed errors and adverse contamination. Albeit PENSE uncovers most of the truly active predictors, the estimate often selects many truly inactive predictors. The issue arises from the elastic net (EN) regularization term in the PENSE objective function, which introduces non-negligible bias and hence cannot lead to a variable selection consistent estimator. Therefore, I propose to replace the elastic net penalty by the adaptive EN penalty which has been shown to lead to variable selection consistent estimators when combined with the LS-loss.

The adaptive EN, as defined in (2.15), combines the advantages of the adaptive LASSO penalty (Zou 2006) and the elastic net penalty (Zou and Zhang 2009). The adaptive LASSO leverages information from a preliminary regression estimate, $\tilde{\boldsymbol{\beta}}$, to penalize predictors with initially "small" coefficient values more heavily than predictors with initially "large" coefficients. This has two major advantages over the non-adaptive EN penalty: (i) the bias for large coefficients is reduced and (ii) variable selection is improved by reducing the number of false positives. Compared to adaptive LASSO, the L_2 term in the adaptive EN improves stability of the estimator in presence of highly correlated predictors (Zou and Zhang 2009).

In this chapter I introduce adaptive PENSE by combining the robust S-loss and the adaptive EN penalty. I state its theoretical properties and show that the adaptive EN

penalty leads to more reliable variable selection than what can be achieved by PENSE. Furthermore, numerical experiments showcase the improved variable selection performance over PENSE while retaining similar predictive power and demonstrates that adaptive PENSE performs better than other variable selection consistent estimators under contamination. The improved variable selection is an important feature for practical applications. I revisit a biomarker discovery study from Cohen Freue et al. (2019) to highlight the utility of the adaptive PENSE estimator.

4.1 Method

The adaptive PENSE estimator is defined by a regularized objective function which combines the robust S-loss and the adaptive EN penalty. The adaptive EN penalty (2.15) is similar to the EN penalty except that the L_1 penalty applied to parameter β_j is scaled by penalty loading ω_j, raised to the power of $\zeta > 0$. For adaptive PENSE, these loadings are set to the reciprocal values of an initial PENSE slope estimate, $\tilde{\boldsymbol{\beta}}^{(\lambda_\text{S},\alpha_\text{S})}$. The objective function for adaptive PENSE is given by

$$\mathscr{O}_\text{AS}(\mu, \boldsymbol{\beta}; \lambda_\text{AS}, \alpha_\text{AS}, \zeta, \boldsymbol{\omega}) = \mathscr{L}_\text{S}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}) + \lambda_\text{AS}\Phi_\text{AN}(\boldsymbol{\beta}; \boldsymbol{\omega}, \alpha_\text{AS}, \zeta) \tag{4.1}$$

with $\omega_j = 1/\tilde{\beta}_j^{(\lambda_\text{S},\alpha_\text{S})}$, $j = 1, \ldots, p$. Minimizers of the adaptive PENSE objective function are denoted by $\hat{\boldsymbol{\theta}}^{(\lambda_\text{AS},\alpha_\text{AS},\zeta,\boldsymbol{\omega})} = \arg\min_{\mu,\boldsymbol{\beta}} \mathscr{O}_\text{AS}(\mu, \boldsymbol{\beta}; \lambda_\text{AS}, \alpha_\text{AS}, \zeta, \boldsymbol{\omega})$. The hyper-parameters are omitted if not pertinent to the argument or obvious from the context.

The interpretations of hyper-parameters λ_AS and α_AS are identical to interpretations of hyper-parameters λ_S and α_S for PENSE, i.e., they control the amount of penalization and the balance between the L_1/L_2 penalties, respectively. The exponent in the predictor-specific regularization, hyper-parameter ζ, is less intuitive. In general, a larger ζ leads to more reliance on the initial estimate $\tilde{\boldsymbol{\beta}}^{(\lambda_\text{S},\alpha_\text{S})}$ for variable selection. A small preliminary coefficient estimate $|\tilde{\beta}_j|$ leads to a larger penalty loading ω_j. With ζ large, this large penalty loading is further amplified, heavily penalizing predictor j which is in turn likely omitted from the active set. Therefore, if ζ is large, only predictors with the very large preliminary coefficient estimates are likely to be selected.

Predictors with a preliminary coefficient estimate of 0 remain inactive after adaptive PENSE. In the formulation of the adaptive EN penalty, these predictors have infinite penalization because $\alpha_\text{AS}\lambda_\text{AS} > 0$ is required. Therefore, these coefficients necessarily stay 0.

When computing the adaptive PENSE estimate according to (4.1), only predictors in the preliminary active set $\mathscr{A}(\tilde{\boldsymbol{\beta}}^{(\lambda_S,\alpha_S)}) = \left\{j : \tilde{\beta}_j^{(\lambda_S,\alpha_S)} \neq 0\right\}$ are considered. While irrelevant for theoretical properties of variable selection performance of adaptive PENSE, the absorbing state at 0 can in practice deteriorate variable selection performance, but at the same time improve computational speed by reducing the complexity of the problem. As an alternative Zou and Hastie (2005) suggest replacing zero coefficients with a very small value ϵ by adjusting the penalty loadings to $\omega_j = 1/\max(\epsilon, |\tilde{\beta}_j^{(\lambda_S,\alpha_S)}|)$. Another way of evading the absorbing state is to use a preliminary estimate with almost surely non-zero coefficients, for example the PENSE-Ridge (i.e., $\alpha_S = 0$). For adaptive PENSE, empirical results suggest that an initial PENSE-Ridge estimate leads to good results and has computational advantages over PENSE estimates with $\alpha_S > 0$.

Finding minima of adaptive PENSE's non-convex objective function is as difficult as for PENSE. The challenge, however, is further elevated by the larger number of hyper-parameters needed for the adaptive PENSE.

4.1.1 Hyper-Parameter Selection

Computing an adaptive PENSE estimate for given values of the hyper-parameters involves two expensive non-convex optimizations: first compute the PENSE estimate $\tilde{\boldsymbol{\theta}}^{(\lambda_S,\alpha_S)}$, then the adaptive PENSE estimate $\hat{\boldsymbol{\theta}}^{(\lambda_{AS},\alpha_{AS},\zeta,\boldsymbol{\omega})}$. An exhaustive hyper-parameter search for adaptive PENSE would in the first stage compute PENSE on a 2-dimensional grid of values for λ_S and α_S. In the second stage, adaptive PENSE is computed on a 3-dimensional grid of values for λ_{AS}, α_{AS}, and ζ, trying every PENSE estimate computed in the first stage. Performing an exhaustive search in this large space is obviously infeasible in practice.

There are several ways to restrict this extensive search. Instead of using every PENSE estimate from the first stage, the search space can be reduced by only considering the "best" PENSE estimate among all PENSE estimates with $\alpha_S = \alpha_{AS}$. A further simplification is to fix the preliminary estimate at the best overall PENSE estimate while still performing a full hyper-parameter search for the adaptive PENSE estimate. For the adaptive LS-EN estimator, Zou and Zhang (2009) propose an even more restricted search. The authors suggest to first select hyper-parameters for the preliminary LS-EN estimate, denoted by α^* and λ^*. For the adaptive LS-EN estimate the authors then only search over the restricted set $\{(\alpha, \lambda)\colon \lambda\frac{1-\alpha}{2} = \lambda^*\frac{1-\alpha^*}{2}\}$, fixing the L_2 penalization in the adaptive EN penalty to the same level as selected for the preliminary LS-EN estimate. This could be translated to the adaptive PENSE estimator by fixing α_{AS} in the second stage to the same value as α_S in the

best overall PENSE estimate.

A different approach to constrain the computational burden of the hyper-parameter search is to compute only the PENSE-Ridge (i.e., $\alpha_{\text{S}} = 0$) in the first stage. This has two advantages: (i) reducing the risk of false negatives in the model selected by adaptive PENSE because the preliminary active set contains all predictors, and (ii) the PENSE-Ridge estimate is faster to compute than PENSE estimates with $\alpha_{\text{S}} > 0$. Although this decreases the computational burden of the first stage considerably, the search in the second stage cannot be restricted and a full 3-dimensional hyper-parameter search is necessary. Empirical results in Section 4.4.1 favor the use of PENSE-Ridge in most applications.

4.2 Statistical Theory

In this section I establish theoretical properties of the adaptive PENSE estimator $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}^0 \in \mathbb{R}^{p+1}$, defined as the global minimum of the adaptive PENSE objective function

$$\hat{\boldsymbol{\theta}} = \underset{\mu,\boldsymbol{\beta}}{\arg\min}\, \mathscr{O}_{\text{AS}}(\mu, \boldsymbol{\beta}; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \boldsymbol{\omega}) \tag{4.2}$$

where $\omega_j = 1/\tilde{\beta}_j^{(\lambda_{\text{S}},\alpha_{\text{S}})}$, $j = 1, \ldots, p$ is determined from an initial PENSE estimate. All hyper-parameters $\lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \alpha_{\text{S}}, \lambda_{\text{S},n}$ are chosen independently of the sample, but $\lambda_{\text{AS},n}$ and $\lambda_{\text{S},n}$ need to decrease according to the number of observations n.

The following asymptotic properties hold under the same general conditions [A1] –[A3] as given for PENSE in Section 3.3. To ease notation and without loss of generality, I assume that the first s components of $\boldsymbol{\beta}^0$ are non-zero (i.e., $\mathscr{A}(\boldsymbol{\beta}^0) = \{1, \ldots, s\}$). The leading non-zero components of the true coefficient vector are denoted by $\boldsymbol{\beta}^0_{\text{I}}$ while the trailing $p - s$ components are denoted by $\boldsymbol{\beta}^0_{\text{II}}$ with $\boldsymbol{\beta}^0_{\text{II}} = \mathbf{0}_{p-s}$.

Proposition 2. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies* (2.2)*. Under assumptions [A1] and [A2] and if $\lambda_{S,n} \to 0$ as well as $\lambda_{AS,n} \to 0$, the adaptive PENSE estimator $\hat{\boldsymbol{\theta}}$ as defined in* (4.2)*, is a strongly consistent estimator of the true regression parameter $\boldsymbol{\theta}^0$: $\hat{\boldsymbol{\theta}} \xrightarrow{a.s.} \boldsymbol{\theta}^0$.*

Noting that the level of L_2 penalization given by $\lambda_{\text{AS},n}\frac{1-\alpha_{\text{AS}}}{2}$ converges deterministically to 0 due to the condition that $\lambda_{\text{AS},n} \to 0$, the proof of strong consistency of adaptive PENSE is otherwise identical to the proof of strong consistency of adaptive MM-LASSO given in Smucler and Yohai (2017) and hence omitted. An important result for the following variable

selection properties is the speed of convergence of the adaptive PENSE estimator, proven in Appendix B.2.2.

Theorem 3. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies (2.2). Under regularity conditions [A1]–[A3] and if $\lambda_{S,n} \to 0$ and $\lambda_{AS,n} = O(1/\sqrt{n})$, the adaptive PENSE estimator $\hat{\boldsymbol{\theta}}$ as defined in (4.2), is a root-n consistent estimator of the true parameter vector $\boldsymbol{\theta}^0$: $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\| = O_p(1/\sqrt{n})$.*

The results so far show that adaptive PENSE theoretically performs as well as PENSE. The adaptive penalty, however, gives rise to an important additional property of adaptive PENSE: variable selection consistency. The following theorem which is proven in Appendix B.2.3 shows that under conditions [A1]–[A3], adaptive PENSE is able to recover the truly active predictors with high probability.

Theorem 4. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies (2.2). Under regularity conditions [A1]–[A3], and if (1) $\lambda_{S,n} = O(1/\sqrt{n})$, (2) $\lambda_{AS,n} = O(1/\sqrt{n})$, (3) $\alpha_{AS}\lambda_{AS,n} n^{\zeta/2} \to \infty$, the adaptive PENSE estimator, $\hat{\boldsymbol{\theta}} = (\hat{\mu}, \hat{\boldsymbol{\beta}})$ as defined in (4.2), is variable selection consistent:*

$$\mathbb{P}\left(\hat{\boldsymbol{\beta}}_{\text{II}} = \mathbf{0}_{p-s}\right) \to 1 \quad \textit{for } n \to \infty.$$

It should be noted that conditions (2) and (3) in the theorem imply that α_{AS} and ζ must be greater than 0. Furthermore, condition (3) is guaranteed to be satisfied for $\zeta > 1$. Using variable selection consistency of adaptive PENSE, it is possible to determine the asymptotic distribution of the estimator of the truly active parameters.

Theorem 5. *Under the same conditions as for Theorem 4 as well as $\sqrt{n}\lambda_{AS,n} \to 0$ the asymptotic distribution of the truly active coefficients of the adaptive PENSE estimator, $\hat{\boldsymbol{\beta}}_{\text{I}}$, is*

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{\text{I}} - \boldsymbol{\beta}_{\text{I}}^0\right) \xrightarrow{\text{d}} N_s\left(\mathbf{0}_s, \sigma_{\text{M}}^2(\mathcal{U})\frac{a(\rho, F_0)}{b(\rho, F_0)^2}\boldsymbol{\Sigma}_{\text{I}}^{-1}\right) \quad \textit{for } n \to \infty.$$

Here, $\sigma_M(\mathcal{U})$ is the population M-scale of the true residuals,

$$\sigma_M(\mathcal{U}) = \inf\left\{s > 0 \colon \mathbb{E}_{F_0}\left[\rho(\mathcal{U}/s)\right] \leq \delta\right\},$$

$a(\rho, F_0) = \mathbb{E}_{F_0}\left[\rho'\left(\mathcal{U}/\sigma_M(\mathcal{U})\right)^2\right]$, $b(\rho, F_0) = \mathbb{E}_{F_0}\left[\rho''\left(\mathcal{U}/\sigma_M(\mathcal{U})\right)\right]$, and $\boldsymbol{\Sigma}_{\text{I}}$ is the covariance matrix of the truly active predictors, $\boldsymbol{\mathcal{X}}_{\text{I}}$.

Together, Theorems 4 and 5 imply that the adaptive PENSE estimator has the same asymptotic properties as if the true model would be known in advance, under fairly mild conditions on the distribution of the predictors and the error term. By the asymptotic nature of these results, they are not immediately transferable to finite samples, especially if the number of predictors is large and the sample size comparatively small. These results are nevertheless useful because they underscore that a large number of irrelevant predictors does not have an undue effect on the accuracy of the estimates; the decisive factor is the number of truly relevant predictors. For practice even more important, these asymptotic results allow for simple comparison of the properties of adaptive PENSE to other competing methods and to understand under what circumstances adaptive PENSE may be preferable. For example, similar results as for adaptive PENSE are obtained for adaptive MM-LASSO in Smucler and Yohai (2017), but their results are contingent on a good estimate of the residual scale. Distinguishing the results in Theorems 4 and 5 from previous work is that the oracle property for the adaptive PENSE estimate can be obtained without prior knowledge of the residual scale, even under very heavy tailed errors.

The scaling factor $\sigma_{\mathrm{M}}^2(\mathcal{U})\frac{a(\rho,F_0)}{b(\rho,F_0)^2}$ in the covariance matrix of the asymptotic Normal distribution of the adaptive PENSE estimator is evidence the adaptive PENSE estimator cannot simultaneously achieve high robustness and high efficiency; the larger δ in the definition of the S-loss, the lower the asymptotic efficiency. The heavier the tail of the error distribution, however, the less severe the loss of efficiency compared to the adaptive MM-LASSO or the adaptive LS-EN. For central stable distributions (Mandelbrot 1960) with stability parameter less than 1.5, for example, the efficiency of adaptive PENSE with $\delta \leq 1/3$, relative to adaptive MM-LASSO, is at least 88%. For adaptive MM-LASSO to achieve higher efficiency than adaptive PENSE the M-loss must be tuned for the specific error distribution. More importantly, however, for the tuning to improve efficiency in finite samples, the residual scale estimate must be close to $\sigma_{\mathrm{M}}(\mathcal{U})$ which is very difficult to achieve in finite samples. Chapter 5 discusses these difficulties in more detail.

The growth rates of $\lambda_{\mathrm{S},n}$ and $\lambda_{\mathrm{AS},n}$ are important to achieve consistency in parameter estimation and variable selection. In practice, however, the hyper-parameters are usually chosen in a data-driven way and hence these growth conditions are almost impossible to enforce or check. The empirical results in Section 4.4 underline that perfect variable selection is very difficult to achieve in finite samples with data-driven hyper-parameter search. Nevertheless, adaptive PENSE shows better variable selection performance than other estimators in challenging problems.

4.3 Robustness Properties

Adaptive PENSE enjoys similar robustness properties as PENSE. The finite-sample breakdown point (FBP) of adaptive PENSE is at least as large as the FBP of the preliminary PENSE estimate. Theorem 2 establishes the breakdown point of the preliminary PENSE estimate is close to δ, where δ is as defined in (2.8) for the S-loss of the preliminary PENSE estimate. If the same δ is used for the adaptive PENSE estimator, it also achieves a breakdown point close to δ, as per the following theorem.

Theorem 6. *For a sample $\mathscr{Z} = \{(y_i, \mathbf{x}_i) \colon i = 0, \ldots, n\}$ of size n, let $m(\delta) \in \mathbb{N}$ be the largest integer strictly smaller than $n \min(\delta, 1 - \delta)$, where δ is as defined in (2.8) for the S-loss of the preliminary PENSE estimate and the S-loss of the adaptive PENSE estimator. Then, for a fixed hyper-parameters $\lambda_S > 0, \lambda_{AS} > 0$ and $\alpha_S, \alpha_{AS} \in [0, 1]$, the breakdown point (2.7) of the adaptive PENSE estimator, $\epsilon^* \left(\hat{\boldsymbol{\theta}}; \mathscr{Z}\right)$, satisfies the following inequalities:*

$$\frac{m(\delta)}{n} \leq \epsilon^* \left(\hat{\boldsymbol{\theta}}; \mathscr{Z}\right) \leq \delta \,.$$

Noting that the preliminary estimate $\tilde{\boldsymbol{\theta}}$ remains bounded by Theorem 2 and hence every coefficient is penalized, the proof is identical to the proof of the FBP of PENSE which is given in Appendix B.1.

4.3.1 Robustness of Variable Selection

In the presence of certain contamination in the predictors, the adaptive EN penalty brings an important advantage over non-adaptive penalties. For PENSE, the smallest penalization level such that $\boldsymbol{\beta} = \mathbf{0}_p$ is a local optimum, as given in (3.5), reveals that a single very large value in a predictor, paired with a non-outlying residual, leads to the explosion of $\widetilde{\lambda}_{\mathrm{AS}}$. Consider the case where predictor j is truly inactive and observation i has an unusually large value for predictor j, i.e., x_{ij} is contaminated. Since predictor j is truly inactive, the response y_i is unaffected by this contamination. From the subgradient of the PENSE objective function at $\boldsymbol{\beta} = \mathbf{0}_p$,

$$\boldsymbol{\nabla}_{\boldsymbol{\beta}} \mathscr{O}_{\mathrm{S}}(\mu, \boldsymbol{\beta}; \alpha_{\mathrm{S}}, \lambda_{\mathrm{S}}) \Big|_{\boldsymbol{\beta}=\mathbf{0}_p} = -\frac{1}{n} \sum_{i=1}^{n} w_i^2 (\mathbf{y} - \mu)\,(y_i - \mu)\,\mathbf{x}_i + \lambda_{\mathrm{S}}[-\alpha_{\mathrm{S}}; \alpha_{\mathrm{S}}],$$

it can be seen that direction j will dominate the gradient, as long as the response y_i is not otherwise contaminated (or exactly fitted by the intercept-only model). Hence, this single

aberrant value in irrelevant predictor j leads to this predictor being the first to enter the model, wrongly suggesting that this predictor is likely relevant.

Standardizing the data beforehand to transform all predictors to the same scale does not mitigate the problem as robust scale estimates would be unaffected by this single contaminated value. A non-robust scale estimate would help to alleviate effects of this particular contamination but would make the regression estimate susceptible to most other forms of contamination. For this reason the classical LS-EN estimator is unaffected by these leverage points when standardizing the predictors by their sample standard deviation.

Inspecting the effects of these leverage points in inactive predictors on PENSE also highlights that the estimated coefficients remain small. Similar to non-regularized estimators, as long as the linear model holds, extremely large values in the predictors actually aid the estimation. These "good" leverage points are highly informative about the true model and force the coefficient value to be close to the true value. In the case where the predictor with these extreme values is truly inactive, the coefficient estimate is forced towards 0. In fact, as $x_{ij} \to \infty$ the estimated coefficient value approaches the true value $\tilde{\beta}_j \to \tilde{\beta}_j^0 = 0$, but it will never be exactly 0 because the predictor eludes the grips of the EN penalty.

Leveraging a preliminary PENSE estimate gives a distinct advantage to adaptive PENSE. Given that the coefficient estimate for the affected predictor is likely small, the penalty loading in adaptive PENSE is very large. This leads to adaptive PENSE most probably screening out these spuriously included predictors, as also showcased in the numerical experiments in Section 4.4.2 Therefore, adaptive PENSE overall has not only theoretically better variable selection properties, but variable selection is also more robust.

4.4 Numerical Experiments

Adaptive PENSE enjoys many important theoretical properties as the sample size increases and hyper-parameter λ_{AS} decreases accordingly. How these properties translate to finite samples and different contamination is not answered by the theory. As with PENSE, the effects of contamination are bounded by theoretical results, but the magnitude is unknown in practice. Continuing the experiments in Section 3.6, the numerical studies presented in this section showcase the benefits of adaptive PENSE in practice.

Additionally to the estimators considered in Section 3.6, adaptive PENSE is compared to several other estimators possessing the oracle property in one or more scenarios considered. Under the same conditions as adaptive PENSE, adaptive MM-LASSO can recover the true

model with high probability even in scenarios where the error distribution has infinite variance, making it a suitable method in the scenarios considered here. Adaptive PENSE and the preliminary PENSE estimate are both tuned to a breakdown point of 33%, while adaptive MM-LASSO chooses the breakdown point automatically between 25–50% based on the degrees of freedom estimated by the S-Ridge (Smucler and Yohai 2017). Hyper-parameters for adaptive PENSE and adaptive MM-LASSO are selected via cross-validation to minimize the estimated τ-size of the prediction error as defined in (3.8). The hyper-parameter ζ for adaptive PENSE is chosen via CV from $\zeta \in \{1, 2\}$. The grid for α_{AS} and λ_{AS} is chosen as in Section 3.6.

The highly robust adaptive PENSE and adaptive MM-LASSO are compared to two other estimators which possess the oracle property, at least for Normal errors. I-LAMM (Fan et al. 2018) with Huber's loss function is also designed for error distributions more heavy-tailed than the Normal, with strong theoretical guarantees even for finite samples, but does require the variance to be finite. For the numerical experiments here, I-LAMM is computed with the methods available in the R package from `https://github.com/XiaoouPan/ILAMM` using the L_1 penalty and default settings. Hyper-parameters are selected via 5-fold cross-validation by the procedure `cvNcvxHuberReg`, with the modification of using the mean absolute prediction error (MAPE) as scale metric to improve performance under heavy tailed error distributions. Adaptive LS-EN, using LS-Ridge as preliminary estimate, is computed by the `glmnet` package in R, with 5-fold CV to select the hyper-parameters minimizing the MAPE.

4.4.1 Preliminary Estimate for Adaptive PENSE

Adaptive PENSE relies on a preliminary PENSE estimate, but theoretical results do not provide guidance on which hyper-parameters are appropriate to compute the preliminary estimate. As outlined in Section 4.1.1, a comprehensive search for all five hyper-parameters is infeasible.

The main goal of adaptive PENSE is to improve variable selection over PENSE while retaining good prediction performance. Figure 4.1 compares two different preliminary PENSE estimates: (i) PENSE (Ridge) computed for $\alpha_{\text{S}} = 0$ with λ_{S} selected via 5-fold CV and (ii) PENSE (CV) with α_{S} and λ_{S} selected via 5-fold CV (this is the PENSE estimate shown in Section 3.6). The plots show the change in sensitivity and specificity of adaptive PENSE compared to the PENSE estimate in percentage points, with the dots representing the median and the error bars extending from the 25% to the 75% quantile.

As expected, specificity of adaptive PENSE is higher than that of PENSE, regardless of the preliminary estimate. In particular when using PENSE (CV) as preliminary estimate, specificity must be at least as high as for PENSE as any predictor excluded by PENSE will necessarily also be excluded by adaptive PENSE. Therefore, leveraging the PENSE (CV) estimate leads to slightly higher specificity than if using PENSE (Ridge). At the same time, adaptive PENSE derived from PENSE (CV) identifies fewer truly relevant predictors because it can only select from those predictors previously selected by PENSE. Using PENSE (Ridge), on the other hand, all predictors are considered when computing adaptive PENSE, and hence the drop in sensitivity from PENSE is more moderate and in many scenarios sensitivity of adaptive PENSE is even higher than sensitivity of PENSE.

It appears as if the benefits of adaptive PENSE decrease as more predictors are available, but it needs to be noted that specificity of PENSE is already quite high in these settings, leaving less room for improvements. In these higher-dimensional problems, PENSE and other regularized estimators have more difficulty identifying the relevant predictors. While adaptive PENSE in general reduces sensitivity even further, leveraging PENSE (Ridge) often leads to an estimate with higher sensitivity in high-dimensional settings.

In terms of prediction performance, adaptive PENSE leads to similar performance as PENSE, albeit slightly reduced. Basing adaptive PENSE on PENSE (CV) tends to decrease prediction performance in the majority of situations as shown in Figure C.7 in the appendix. Prediction performance of adaptive PENSE with PENSE (Ridge) as the preliminary estimate, on the other hand, is not substantially different from PENSE.

Overall, adaptive PENSE based on the PENSE (Ridge) preliminary estimate improves specificity without sacrificing as much sensitivity as if using PENSE (CV). Leveraging PENSE (Ridge) can even be beneficial for sensitivity in high dimensions and does not impede prediction performance of the estimate. In applications where the costs associated with including irrelevant predictors is prohibitive, PENSE (CV) may be the more appropriate preliminary estimate for adaptive PENSE. In general, however, adaptive PENSE based on PENSE (Ridge) leads to an overall more substantial improvement of variable selection properties with similar prediction performance as PENSE. In subsequent numerical experiments, adaptive PENSE is therefore reported with PENSE (Ridge) as preliminary estimate.

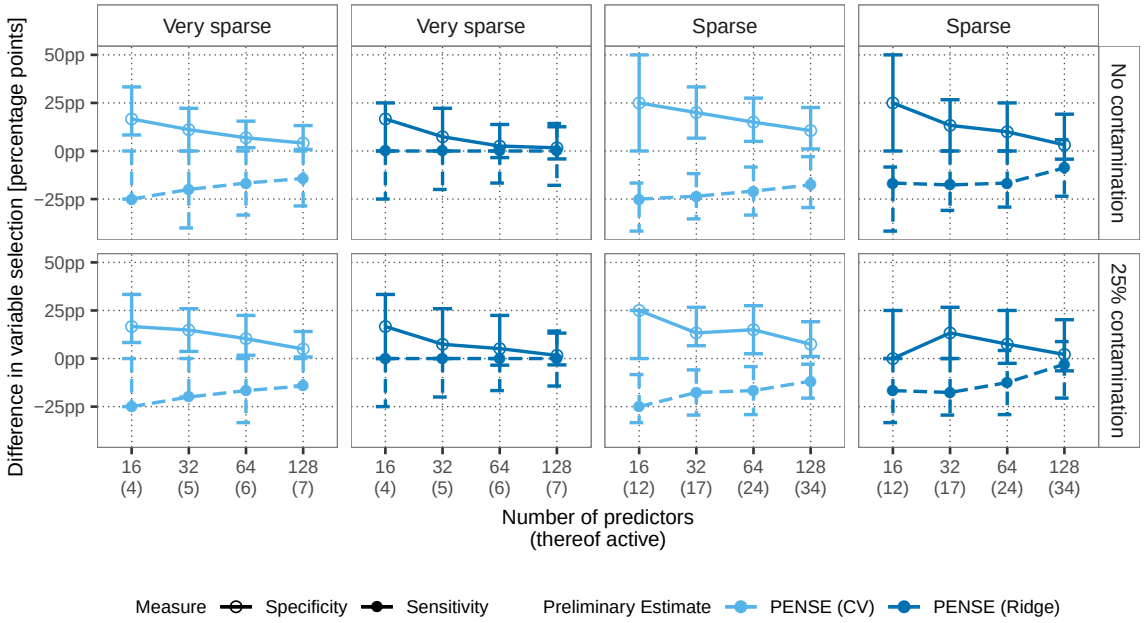


Figure 4.1: Comparison of variable selection performance of adaptive PENSE using different preliminary estimates. Data is simulated according to schemes *VS1-** in panels on the left and *MS1-** in panels on the right, with $n = 100$ and 25% variance explained by the true model. Results for "no contamination" (top) show the median and inter-quartile range over 100 replications, while results on the bottom summarize 50 replications for each of 6 scenarios with different contamination settings.

4.4.2 Effects of Good Leverage Points

Combining the robust S-loss with the adaptive EN penalty promises more robust variable selection in the presence of good leverage points as detailed in Section 4.3.1. To support this statement with empirical results, data is generated according to scheme *MS1-MH(–, k_l)* with $p = 32$ predictors and $n = 100$ observations with adapted contamination model. All 100 response values are generated according to the true model, but in 10% of observations, some predictor values are contaminated by

$$\tilde{x}_{i,15+i} = x_{i,15+i} k_1 \frac{\max_{i'=1,\dots,n} d_i^2}{d_{15+i}^2} \qquad \text{for } i = 1, \dots, 10 \tag{4.3}$$

with d_i^2 the squared Mahalanobis distance of observation i, relative to the 10 contaminated predictors, as in (A.3). In other words, each of the first 10 observations has a single predictor with unusually large value, with the severity of leverage controlled by parameter k_1. The first 17 predictors are truly active; hence this contamination model introduces leverage points in 2 truly active predictors and 8 truly inactive predictors.

Results are shown in Figure 4.2, underlining that PENSE estimates are considerably affected by these "good" leverage points. Sensitivity and specificity are calculated separately for contaminated (top) and uncontaminated predictors (bottom). All estimates select the truly active predictors with contamination in the vast majority of replications, regardless of the severity of leverage introduced. As predicted, PENSE almost always selects all truly irrelevant predictors with contamination. Adaptive PENSE using PENSE with $\alpha = 0$ as preliminary estimate, on the other hand, shows highly consistent variable selection performance over all leverage parameters, k_1. Adaptive PENSE is able to identify most truly active predictors (contaminated or not), while also screening out large parts of the truly inactive predictors. Sensitivity of I-LAMM estimates drops drastically as the severity of the leverage points increases, with specificity increasing in tandem. Therefore, in the presence of very severe leverage points, I-LAMM selects only the contaminated truly active predictors, everything else is excluded from the model. Non-robust (adaptive) LS-EN show fairly good variable selection with high specificity for both contaminated and uncontaminated predictors, but the trajectory of sensitivity follows a similar trajectory as I-LAMM, albeit the decrease is more gradual.

Good leverage points seem to be more helpful for non-robust estimates, up to the point where the leverage becomes too severe and overshadows the other truly active predictors. Adaptive PENSE maintains a high level of sensitivity and specificity for any severity of good leverage points, but compared to non-robust estimators, these variable selection properties also persist in the presence of other contamination.

4.4.3 Overall Effect of Contamination

Adaptive PENSE performs reliably under the presence of good leverage points. Assessing the impact of a greater variety of contamination, adaptive PENSE and other variable selection consistent estimators are computed in the same scenarios as considered in Section 3.6.

Figure 4.3 summarizes the prediction performance for scenarios with $n = 100$ observations. I-LAMM, with Huber's loss and LASSO penalty is not robust towards high leverage points in the predictors but outperforms robust estimators for Normal errors and no contamination. When the error distribution is heavy-tailed or when gross contamination is introduced, predictions from I-LAMM estimates tend to give higher errors than predictions from PENSE or adaptive PENSE. Across all scenarios, adaptive PENSE estimates have very similar predictive power as PENSE estimates, as evident from the results reported in Appendix C.2.1. Adaptive MM-LASSO performs as good as adaptive PENSE in very

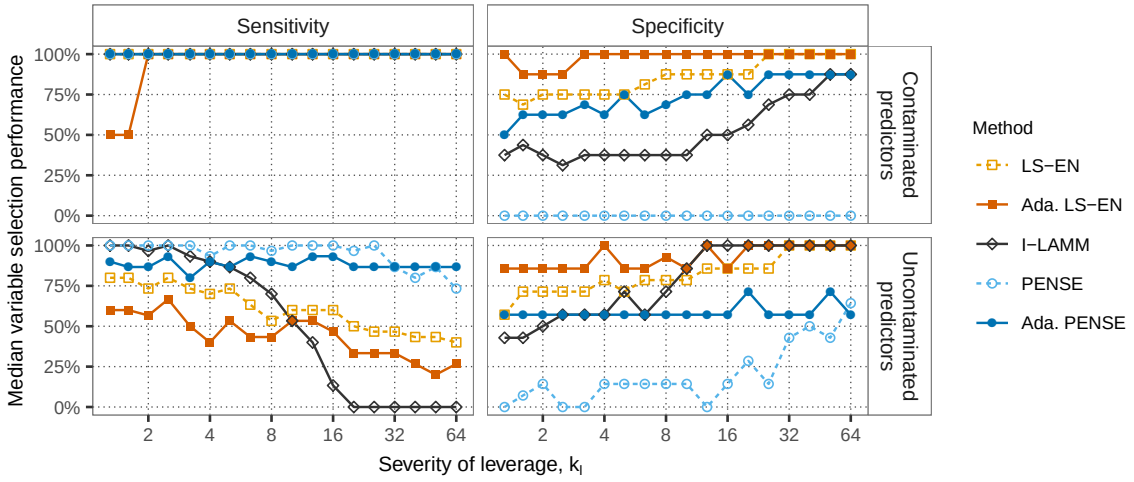
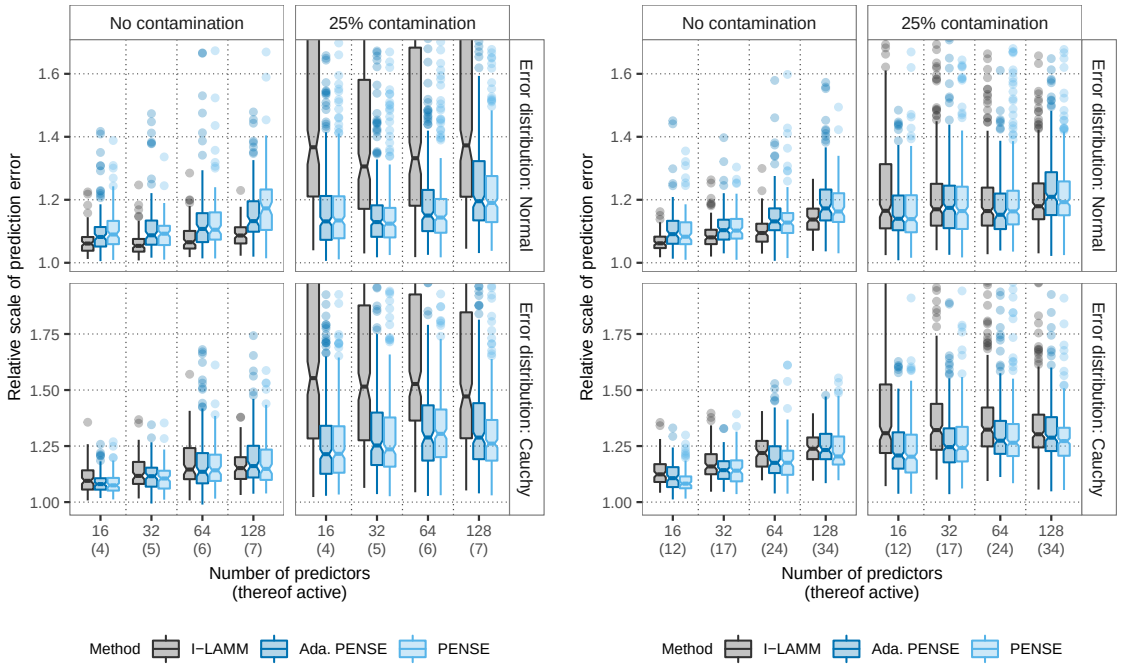


Figure 4.2: Effect of high-leverage points on the sensitivity and specificity of variable selection. Median values over 50 replications of these measures are reported separately for predictors containing contaminated values and predictors free from any contamination. Data is generated according to scheme *MS1-MH** with $p = 32$, $n = 100$, 75% variance explained by the true model and 10% contamination introduced according to (4.3).

sparse scenarios, but more active predictors are better handled by adaptive PENSE. Conclusions for the estimation accuracy reported in Appendix C.2.3 coincide with prediction performance.

Variable selection performance of adaptive PENSE, shown in Figure 4.4, underscores the conclusions from previous experiments. Adaptive PENSE is performing similar to I-LAMM in very sparse scenarios with no contamination, but adaptive PENSE is more robust towards heavy tailed errors and leverage points. Compared to PENSE, adaptive PENSE estimates screen out more truly irrelevant predictors, at the cost of missing some truly relevant ones. Noting that in very sparse scenarios (Figure 4.3(a)), the introduced outliers are more extreme than in sparse scenarios (Figure 4.3(b)), adaptive PENSE has almost the same sensitivity as PENSE under the presence of severe leverage points combined with gross outliers, but adaptive PENSE excludes many more irrelevant predictors. Adaptive MM-LASSO, as shown in Appendix C.2.2, has substantially lower sensitivity than adaptive PENSE in the vast majority of scenarios. Compared to other variable selection consistent estimators, adaptive PENSE tends to retain more truly active predictors while still screening out most of the irrelevant predictors. Variable selection properties of adaptive PENSE are less affected by outliers and heavy-tailed errors than I-LAMM estimates or MM-LASSO estimates. Adaptive PENSE strikes a balance between high specificity achieved by LASSO-type estimators and high sensitivity of PENSE estimates. This is especially useful in applications where a



(a) Very sparse scenarios *VS1-LT** and *VS1-HT**.

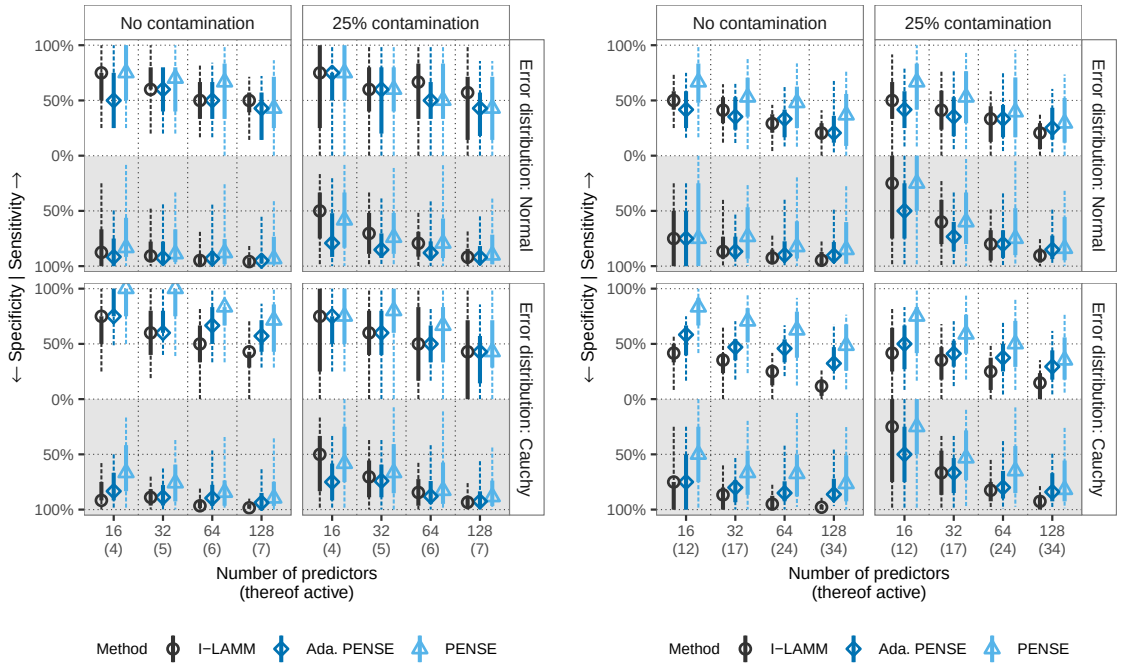
(b) Sparse scenarios *MS1-LT** and *MS1-HT**.

Figure 4.3: Prediction performance of regression estimates in different scenarios with a sample size of $n = 100$. The horizontal axis in each panel shows the total number of predictors, while the vertical axis in each panel shows the root mean square prediction error (for Normal errors) or the τ scale of the prediction errors.

small number of false negatives can be tolerated at the benefit of substantially reducing the number of true negatives.

4.5 Biomarkers for Cardiac Allograft Vasculopathy

In Cohen Freue et al. (2019) we demonstrate the usefulness of PENSE in clinical biomarker discovery studies. In this application, the overarching goal is to identify a small set of proteins which help to detect whether a patient suffers from cardiac allograft vasculopathy (CAV). CAV is a common complication in patients who received a cardiac transplant. Almost 50% of recipients develop CAV in the years following transplantation (Cohen Freue et al. 2019), accounting for almost 15% of deaths in heart transplant recipients who survived the first year after transplantation (Lin et al. 2013). In clinical practice, transplant recipients are monitored at least annually for the onset of CAV. Diagnostics typically rely on coronary angiography, measuring the narrowing of arteries supplying oxygenated blood



(a) Very sparse scenarios *VS1-LT** and *VS1-HT**.

(b) Sparse scenarios *MS1-LT** and *MS1-HT**.

Figure 4.4: Sensitivity and specificity of regression estimates in different scenarios with a sample size of $n = 100$. The horizontal axis in each panel shows the total number of predictors. The vertical axis in each panel is split in two halves: sensitivity (i.e., the number of correctly identified active predictors) is shown on the top half, and specificity (i.e., the number of correctly identified inactive predictors) is shown downwards with perfect specificity (100%) on the bottom. Solid vertical lines show the range of the inner 50%, while the dashed lines extend from the 5% to the 95% quantile.

to the heart (Schmauss and Weis 2008). Coronary angiography is an invasive procedure prone to complications (Lin et al. 2013). A simple blood test targeting specific proteins in the plasma could potentially reduce the risks to patients substantially and improve health outcomes of heart transplant recipients.

The data used here was first analyzed in Lin et al. (2013) and later in Cohen Freue et al. (2019), comprising information on 37 cardiac transplant recipients. All 37 patients were assessed for CAV by measuring the maximum percentage of diameter stenosis (Max %DS) in the left anterior descending (LAD) artery (Lin et al. 2013). The original proteomic data consists of measurements of hundreds of proteins detected in blood plasma samples from the 37 recipients. Following the analysis in Cohen Freue et al. (2019), I utilizes only the 81 proteins reliably detected across all plasma samples.

The statistical goal is to predict the Max %DS in the LAD through a linear model of the measured protein levels such that only some of the proteins are included in the linear

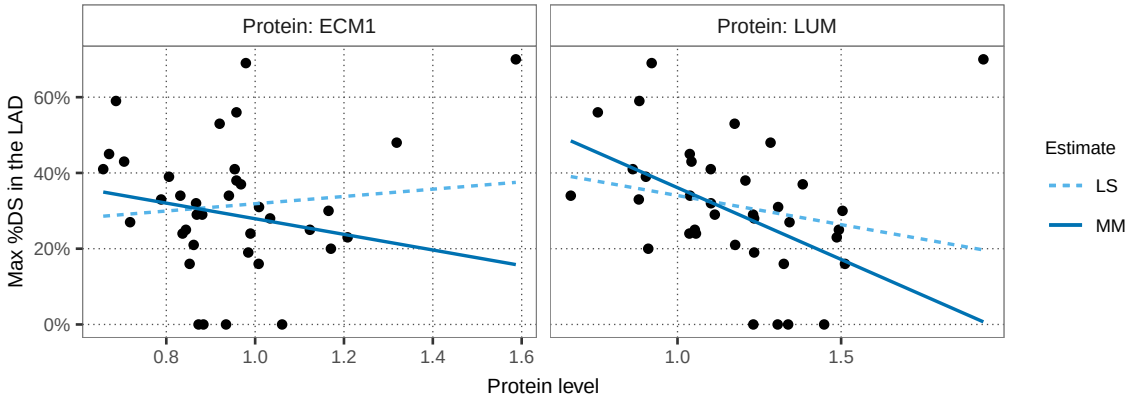


Figure 4.5: Univariate regression estimates for regressing the maximum percentage of diameter stenosis (Max %DS) in the LAD artery on the level of proteins ECM1 and LUM in the CAV case study.

relationship. Limiting the number of relevant proteins is important for a viable blood test, as the costs of a test targeting many proteins would prohibit a wide-spread use.

Exploratory analysis of the data suggests that the measurement of Max %DS in the LAD but also some protein levels contain possibly contaminated values. Figure 4.5, for instance, shows the results of univariate regressions of the response variable on the measured levels of proteins ECM1 and LUM. The robust univariate MM-estimate detects a negative relationship between the protein levels and Max %DS in the LAD vessel in the sample at hand. The classical least squares estimate (LS), on the other hand, estimates a positive relationship between ECM1 and the response variable and a substantially smaller effect of LUM. For both proteins, a few patients with unusually severe narrowing of the LAD combined with a comparatively high abundance of proteins ECM1 and LUM in their blood plasma excessively affect the LS estimate. Several similar instances of contamination in the sample cast doubt on the appropriateness of non-robust methods for identifying relevant proteins and quantifying their effect.

Comparison of the prediction performance of several estimates in the CAV study is done by nested cross-validation. Specifically, the sample of 37 observations is split into 7 CV folds (the "outer" folds). Within each outer fold, an "inner" 7-fold CV is used to select hyper-parameters individually for each estimator. To counter the inherent variability in cross-validation for robust estimators, the inner CV for all estimators is repeated 50 times (see also Chapter 6 for details on repeated CV for PENSE and adaptive PENSE). As in the numerical experiments, (adaptive) PENSE choose hyper-parameters to minimize the τ-size of the prediction error, while other methods minimize the mean absolute prediction

error. With these selected hyper-parameters, the left-out observations from the outer fold are predicted and the scale of the prediction error recorded. The outer CV is replicated 100 times to assess overall prediction performance of the considered estimators in the CAV study.

Results of nested CV are shown in Figure 4.6(a). The difference in prediction performance between the estimates is not very pronounced, but nevertheless noticeable. This is in line with the prediction performances reported in Cohen Freue et al. (2019), albeit results reported here suggest slightly better performance for all estimators because repeating the inner CV leads to more stable hyper-parameter selection. Adaptive PENSE leads on average to better prediction performance than the other methods considered. Adaptive LS-EN performs poorly in the CAV study, much like in the numerical experiments under the presence of contamination. The initial LS-Ridge estimate is likely affected by contamination, and hence "leveraging" this estimate amplifies the effect of contamination. The number of relevant predictors selected varies between CV splits, but in general adaptive PENSE and I-LAMM select far fewer proteins than the other methods.

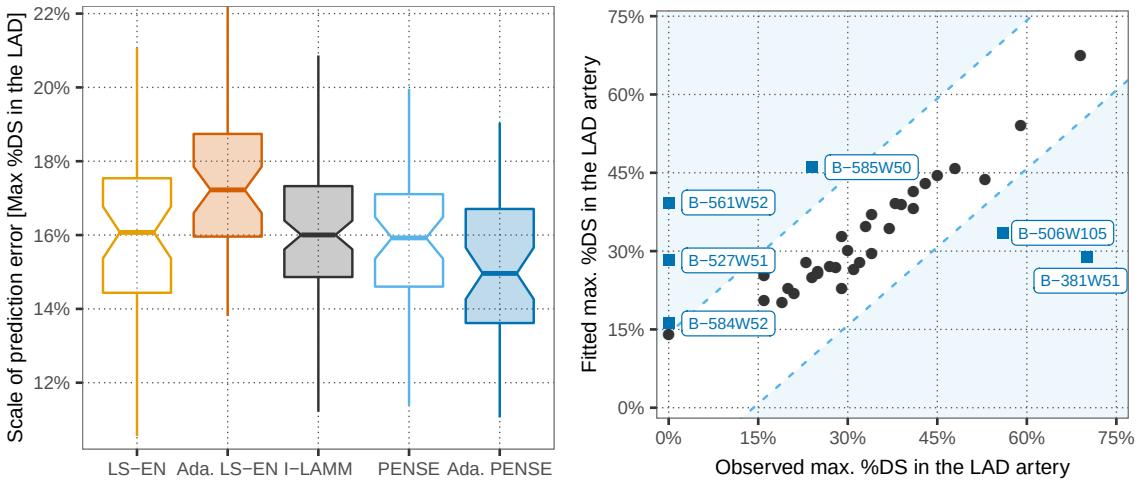
Each method is also applied to the full sample, again using repeated 7-fold CV to select hyper-parameters. For all but LS-EN, the hyper-parameters are not selected to achieve minimum scale of the prediction error, but rather to lead to the most parsimonious model with a scale of the prediction error not substantially worse than the minimum (within 1/2 the standard error of the minimum). For LS-EN, this "half standard error rule" always leads to the empty model, a typical observation with LS-EN under high noise in the response variable. Prediction performance may be similar, but the proteins selected by the different estimates vary substantially. Non-robust LS-EN and adaptive LS-EN select 21 and 20 proteins, respectively, with adaptive LS-EN dropping only a single protein. Similarly, PENSE detects 20 relevant proteins, 13 overlapping with (adaptive) LS-EN. Adaptive PENSE and I-LAMM select the smallest number of proteins among the considered estimators, 14 and 12, respectively, but based on the prediction performance estimated before, the panel identified by adaptive PENSE, listed in Table 4.1, is likely more relevant for predicting CAV. Half of the proteins identified by adaptive PENSE overlap with proteins selected by non-robust methods, but adaptive PENSE detects several novel proteins.

In Cohen Freue et al. (2019), we improve upon the model fitted by PENSE via a subsequent M-step (PENSEM), selecting a total of 15 proteins. The proteins selected by adaptive PENSE, PENSEM, and Lin et al. (2013) are listed in Table 4.1. Three proteins are selected by all three methods, while adaptive PENSE and PENSEM overlap in four additional pro-

teins. Interestingly, adaptive PENSE selects the extracellular matrix proteins ECM1 and LUM, which have been linked to coronary artery disease (Zhao et al. 2016) and formation of new blood vessels (Neve et al. 2014). Lumican (LUM) is also determined relevant in Lin et al. (2013), but ECM1 is selected only by robust estimators, potentially because of contamination highlighted in Figure 4.5 transmogrifies the predominantly negative effect of ECM1 on the response into a positive effect. Adaptive PENSE also detects some novel proteins not previously associated with CAV, most notably Hemopexin (HPX). Hemopexin has been targeted to improve cardiovascular function (Vinchi et al. 2013) and is associated with several inflammatory diseases (Mehta and Reddy 2015).

We show that the PENSEM estimator can lead to improved prediction performance over PENSE and other robust estimators (Cohen Freue et al. 2019). The M-step is supposed to increase efficiency of the initial S-estimator, similar to the idea of MM-LASSO and classical MM-estimators. However, just like MM-LASSO, the M-step for PENSEM hinges on the accuracy of the residual scale estimated by the initial S-estimator. Especially in higher dimensions or if the true error distribution is heavy tailed, however, the scale estimate derived from PENSE or S-Ridge may not be relied upon. This can lead to severe problems for PENSEM and MM-LASSO, as highlighted in the numerical experiments in this and the previous chapter.

Compared to the model fitted by PENSEM, adaptive PENSE detects a stronger signal for fitting and predicting CAV using a smaller panel of proteins. With adaptive PENSE, the maximum percentage diameter stenosis in the LAD vessel can be fitted well as shown in Figure 4.6(b). Additionally, the robust nature of the estimate allows identification of several patients with unusual stenosis. Patients with residuals located in the shaded regions of Figure 4.6(b) are more than two standard errors (estimated by the M-scale of the residuals) away from the diagonal and can be considered outliers. The adaptive PENSE estimate suggests that in six patients the measured Max %DS is suspiciously different from what could be expected based on their proteomic profile. Most of these patients are also flagged by PENSEM as having unusual response values, but more severe and mild stenosis is fitted substantially better by adaptive PENSE than PENSEM. A follow-up measurement using more accurate intravascular ultrasound revealed three patients with initially no stenosis detected, B-584, B-527 and B-561 (initially measured in weeks 51 or 52 after transplant), have indeed developed mild stenosis of the LAD artery of about 16 Max %DS, very close to the values fitted by adaptive PENSE. Adaptive PENSE identifies a small set of proteins leading to superior prediction performance and a better fit to the data than other methods.



(a) Estimated prediction performance.

(b) Observed vs. fitted Max %DS in the LAD from adaptive PENSE.

Figure 4.6: Results of the CAV study: (a) the scale of the prediction error of the maximum percentage of diameter stenosis (Max %DS) for several estimators in the CAV study and (b) observed values versus values fitted by adaptive PENSE. Prediction errors in (a) are estimated via nested 7-fold cross-validation, repeated 100 times. The shaded regions in (b) depict residuals farther than twice the standard error away from the fitted value, indicating unusually large residuals. Hyper-parameters for the adaptive PENSE fit in (b) are selected by 7-fold CV, repeated 100 times.

With the demonstrated robustness of variable selection, adaptive PENSE is an important addition to the toolbox for biomarker discovery.

4.6 Conclusions

The elastic net S-estimator, PENSE, introduced in Chapter 3 has highly competitive prediction performance even under the presence of adverse contamination. Furthermore, PENSE is demonstrated to identify the vast majority of truly relevant predictors, but PENSE estimates often wrongly include a very high number of irrelevant predictors. The adaptive elastic net S-estimator, adaptive PENSE, is devised out of the need of controlling the excessive rate of false discoveries made by PENSE estimates.

Adaptive PENSE is shown to possess two important asymptotic properties missing from PENSE: variable selection consistency and the oracle property. In Section 4.2 it is proved that adaptive PENSE estimators are variable selection consistent even in settings where the error distribution does not have finite variance. Variable selection consistency is the key ingredient for showing that adaptive PENSE estimates of the coefficients of truly active

Table 4.1: Proteins identified by adaptive PENSE to predict Max %DS in the LAD artery, compared to proteins selected by other methods.

Gene symbol	Protein name	Adaptive PENSE	PENSEM	Lin et al. (2013)
AMBP	Protein AMBP	✓	✓	✓
APOE	Apolipoprotein E	✓	✓	✓
C4B;C4A	Complement C4-B/C4-A	✓	✓	✓
ECM1	Extracellular matrix protein 1	✓	✓	
F2	Prothrombin (Fragment)	✓	✓	
HBA2;HBA1	Hemoglobin alpha-2	✓	✓	
HBD	Hemoglobin subunit delta	✓	✓	
C7	Complement component C7	✓		✓
LUM	Lumican	✓		✓
C1R	Complement C1r subcomponent	✓		
HABP2	Hyaluronan-binding protein 2	✓		
HPX	Hemopexin	✓		
SERPINA3	Alpha-1-antichymotrypsin	✓		
SERPINC1	Antithrombin-III	✓		

predictors as precise as if the truly active predictors were known in advance. Therefore, even in problems with many available predictors, coefficients of the active predictors are accurately estimated.

The adaptive elastic net penalty also improves robustness of variable selection as outlined in Section 4.3.1 and demonstrated numerically in Section 4.4.2. Contamination of inactive predictors in observations which follow the true linear model causes PENSE estimates to wrongly include these predictors in the model. This leads to a breakdown of variable selection of PENSE, where contamination with leverage points severely degrades specificity. The robustness of the S-loss, however, ensures that the coefficient estimates for truly inactive predictors with excessive leverage remain small. By leveraging a robust PENSE estimate, adaptive PENSE is able to screen out many of these spuriously selected predictors. Empirical observations suggest PENSE with the Ridge penalty ($\alpha_S = 0$) is the appropriate preliminary estimate in most applications. With the Ridge penalty, PENSE can be computed more efficiently than with a non-smooth penalty ($\alpha_S > 0$) and hyperparameter selection is substantially less demanding. Furthermore, sensitivity decreases only moderately compared to the best PENSE estimate while specificity increases.

Adaptive PENSE's increased robustness towards leverage points is an important property for real-world applications. Section 4.5 revisits a biomarker discovery study with the

goal of identifying proteins in the human blood plasma which help to predict cardiac allograft vasculopathy, a major complication after heart transplantations. In Cohen Freue et al. (2019) we use PENSE and a subsequent M-step to determine a panel of 15 possibly relevant proteins. The M-step proves to be challenging in this application due to the difficulty of estimating the scale of the residuals accurately. When applying adaptive PENSE to the same data set, a slightly different panel of 13 possible relevant proteins is uncovered. The adaptive PENSE estimate leads to superior prediction performance in the study and at the same time fits the data better than competing robust and non-robust methods.

Theoretical results expose the major drawback of regularized S-estimators: substantially lower asymptotic efficiency than regularized M-estimators for light- and moderate-light-tailed error distributions. Empirical results are in line with this observation, although the differences between regularized M- and S-estimators in finite-samples are far less pronounced than suggested by the theory. The following chapter discusses challenges of robustly estimating the residual scale and thereby sheds light on reasons why regularized M-estimators may not provide the gains in efficiency in finite-samples as promised by the theory.

Chapter 5

Residual Scale Estimation

S-estimators of regression are highly robust to aberrant contamination in the data and heavy tailed error distributions. In Chapters 3 and 4 I show that this also holds for PENSE and adaptive PENSE, even in high dimensions. The apparent downside of S-estimators, already discussed in Section 2.2, are their low efficiency under the Normal model. An iconic idea in robust statistics is to follow the S-estimator by an additional M-step (Yohai 1987). The resulting MM-estimator of linear regression inherits the robustness properties from the initial estimator but can be tuned to achieve high efficiency arbitrarily close to the LS-estimator. In their most basic form, MM-estimators are defined by the following sequence of steps:

Step 1 Compute a highly robust and strongly consistent estimate of regression, e.g., the PENSE estimate $\hat{\boldsymbol{\theta}}_{\text{S}}$.

Step 2 Compute the M-scale of the residuals from the estimate fitted in step 1, $\hat{\sigma}_{\text{S}} = \hat{\sigma}_{\text{M}}(\mathbf{y} - \hat{\mu}_{\text{S}} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{S}})$.

Step 3 Using the S-estimate $\hat{\boldsymbol{\theta}}_{\text{S}}$ as initial estimate, find a local minimum $\hat{\boldsymbol{\theta}}_{\text{MM}}$ of

$$\mathscr{L}_{\text{M}}(\mathbf{y}, \mathbf{X}\boldsymbol{\beta} + \mu; \hat{\sigma}_{\text{S}}) = \frac{1}{n}\sum_{i=1}^{n} \rho_{\text{M}}\left(\frac{y_i - \mu - \mathbf{x}_i^{\intercal}\boldsymbol{\beta}}{\hat{\sigma}_{\text{S}}}\right)$$

which improves upon the initial estimate, i.e., $\mathscr{L}_{\text{M}}(\mathbf{y}, \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{MM}} + \hat{\mu}_{\text{MM}}) \leq \mathscr{L}_{\text{M}}(\mathbf{y}, \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{S}} + \hat{\mu}_{\text{S}})$. Here, ρ_{M} is a bounded ρ function according to [R1]–[R3] which is dominated by the ρ function used to compute the M-scale (2.8), i.e., $\rho_{\text{M}}(t) \leq \rho(t)$ for all t.

Evidently, the M-step is computationally cheap, given that only a single starting point

needs to be considered and the objective function is separable over the observations. In Cohen Freue et al. (2019) we adopt this idea to improve upon the PENSE estimate by a subsequent M-step, called PENSEM. Smucler and Yohai (2017) base their MM-LASSO on the same principle, but the execution is slightly different than what is done for PENSEM. These differences highlight some of the challenges translating the idea of MM-estimators to a high dimensional setting using robust regularized estimators.

None of the three steps for computing MM-estimates can be applied to regularized estimators without modification. In step 1, the question is what hyper-parameters should be selected for a robust regularized estimate of regression. For MM-LASSO, the choice is to compute the S-Ridge estimate, i.e., using $\alpha_{\mathrm{S}} = 0$, optimizing the penalization level for prediction performance. PENSEM, on the other hand, uses the PENSE estimate with both α_{S} and λ_{S} optimized for prediction performance. Others propose to not use a regularized estimate in step 1 but an unregularized MM-estimate (Arslan 2016), which is only possible for low-dimensional problems.

Step 3 raises similar questions as step 1 about an appropriate choice of hyper-parameters and whether a local minimum close to the estimate from step 1 is a sensible choice. Due to vastly different scales of the loss functions in the two steps, the penalization level selected in step 1 is in general not a reasonable choice for the M-step. Both MM-LASSO and PENSEM carry out a separate hyper-parameter search for the M-step; MM-LASSO for the penalization level, PENSEM for α and λ. The theoretical results in Yohai (1987) justify using only the consistent and robust estimate from step 1 as starting point for computing the MM-estimate, as the local minimum uncovered has the same asymptotic properties as the global minimum. No such results are available for the regularized M-step, but MM-LASSO and PENSEM nevertheless follow the same principle and do not perform an exhaustive search for good initial estimates to restrain the computational overhead. Despite this leap of faith, both MM-LASSO and PENSEM show an improved efficiency over the initial S-estimate, but not in every setting. Most concerning is the observation that MM-LASSO and PENSEM seem to be much more affected by contamination in some situations.

The main problem why regularized M-steps do not always improve efficiency, and sometimes seemingly break down, is the difficulty posed by step 2. For the M-step to perform as expected under the assumed model, the ρ_{M} function, more specifically the cutoff value, is chosen based on the probabilistic limit of $\hat{\sigma}_{\mathrm{S}}$. This of course requires the assumed model to hold for the majority of observations, but the greater challenge in practical applications is the bias of the estimate $\hat{\sigma}_{\mathrm{S}}$. Even if $\hat{\sigma}_{\mathrm{S}}$ converges almost surely to a fixed limit, the finite-

sample bias may be arbitrarily large. If the bias in finite samples is too large, the chosen ρ_{M} may not deliver the promised gains in efficiency. Especially in higher dimensions, the bias in the residual scale estimate can be unacceptably large.

5.1 The Problem in High Dimensions

Estimating the scale of the residuals in high dimensional linear models is known to be difficult and prone to bias. Already Mammen (1996) paints a bleak picture, showing how fast the bias of the empirical distribution of the residuals in a p-dimensional linear regression model increases with p. The bias in the empirical residuals, if not corrected, translates to a biased scale estimator. The problem is even amplified by the use of regularized and/or robust estimators, but it has only recently been attracting attention (e.g., Fan et al. 2012; Dicker 2014; Chatterjee and Jafarov 2015; Reid et al. 2016; Chen et al. 2018; Tibshirani and Rosset 2019).

Regularized estimators have been studied extensively, but attention was mostly directed at prediction and variable selection performance of these estimators. Perhaps pushing the issue to the sidelines even more, theoretical properties of regularized estimators do not depend on an estimate of the residual scale. With emergence of more literature on post-selection inference, however, residual scale estimation has become more closely investigated.

The review paper by Reid et al. (2016), and the recent proposals by Yu and Bien (2019) and Chen et al. (2018) highlight the numerous challenges in estimating the residual variance using regularized estimators. As already outlined in Fan et al. (2012), residual scale estimation with regularized estimators is impeded by the inherent bias of the estimates. The main sources of the bias are the penalization of the coefficients and data-driven selection of the hyper-parameters with a goal of good prediction. Unfortunately, these two sources of bias work in tandem, accentuating their effect on the residual scale estimate. In high dimensions, predictors spuriously correlated with the response add to the problem. The larger the number of irrelevant predictors, the greater the chances of spurious correlation and hence overfitting the response.

The problem is not restricted to classical, least-squares-based estimators, but affects robust estimators even more. For non-regularized MM-estimators, Maronna and Yohai (2010) show how serious underestimation of the error scale by the M-scale from the residuals of the S-estimate can be. Even in a setting considered low dimensional in this work ($n = 50$ and $p = 15$) the estimate of the error scale is below half of the true error scale almost 50%

of times. These results are for non-regularized MM-estimators and do not account for the impact of regularization.

The down-stream effect of a poor scale estimate on the M-step can be devastating. The M-loss function depends on the boundedness of the ρ_{M} function to protect against gross outliers, while at the same time behaving similarly to the LS-loss for small residuals to ensure efficiency. Considering a severe underestimation of the error scale, $\hat{\sigma}_{\mathrm{S}} \ll \sigma_{\mathcal{U}}$, the scaled residuals $\frac{y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta}}{\hat{\sigma}_{\mathrm{S}}}$ are artificially inflated. This "pushes" many scaled residuals into the bounded region of the ρ_{M} function, treating them as outlying. Therefore, the M-step does not improve efficiency because a large proportion of actually uncontaminated observations are incorrectly down-weighted. Severe overestimation of the error scale, on the other hand, shrinks the scaled residuals towards 0, neutralizing the boundedness of ρ_{M}. In this case, outlying observations are not detected as such and can grossly affect the M-estimate. An inaccurate estimation of the error scale can thus either lead to a decrease in efficiency compared to the initial S-estimator, or even jeopardize the robustness of the M-estimator.

Estimating the residual scale with PENSE suffers from the bias inflicted by the M-scale in addition to the bias introduced by the penalty function and data-driven hyper-parameter selection. As depicted in Figure 5.1 for simulated data, the effects of a poor scale estimate on the subsequent M-step, PENSEM, are worrisome. Firstly, the plots clearly show the prevalence of severe underestimation and overestimation. Underestimation is commonly observed even without contamination, but the scale is often severely overestimated in the presence of contamination, especially when combined with heavy-tailed errors. It is evident that gross under- or overestimation of the residuals scale leads to a degradation of prediction performance of PENSEM, with distressing effects under the presence of contamination. The conclusions are the same for any regularized M-estimator relying on an initial scale estimate in high dimensions: in the majority of cases the M-step improves efficiency and leads to better estimation and prediction, but in an unsettling large number of instances the M-estimator is either less efficient than the initial S-estimator, or worse, seriously affected by contamination.

Without an improved residual scale estimate in high dimensional problems, results from PENSEM or other regularized redescending M-estimators may be unreliable. As an ad-hoc solution for unpenalized MM-estimators, Maronna and Yohai (2010) suggest a multiplicative correction to increase the residual scale estimated from the S-estimate. Smucler and Yohai (2017) use this correction for the MM-LASSO estimator, but empirical results presented here suggest the adjustment does not work well for regularized estimators. The residual

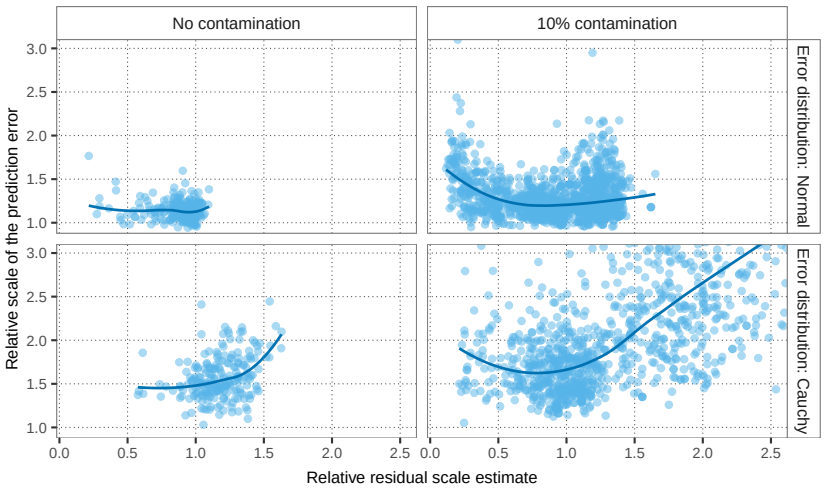


Figure 5.1: Prediction performance of the PENSEM estimate as a function of the residual scale estimated by PENSE. Hyper-parameters are selected via 5-fold CV. The residual scale estimate on the horizontal axis and the scale of the prediction error are reported relative to the true M-scale of the residuals. Data is generated according to scheme *VS1-LT** (top) and *VS1-HT** (bottom) for $n = 100$ and $p \in \{50, 100\}$. The true model explains 83% of the variation and results under contamination (right) consider scenarios with 10 different vertical outlier positions and $k_l = 6$.

scale estimate is often overestimating the true scale as suggested by the simulation results reported before; further inflating the scale estimate in these situations exacerbates the problem. For non-robust estimators, several strategies for correcting the bias in the scale estimate have been proposed in the literature. The majority of these proposals is based on the idea of splitting the data into non-overlapping chunks.

5.2 Data-Splitting Strategies

One of the driving forces behind the bias in regularized estimators is data-driven hyper-parameter selection. With penalization leading to an underestimation of the coefficients, cross-validation or similar strategies to give good prediction performance compensate for the biased coefficients by selecting a penalization level which is too small to screen out spuriously correlated predictors. The regularized estimate computed on the entire data set with this small penalization level will typically include some of these irrelevant predictors. Just by chance of observing these spuriously correlated predictors, the fitted model explains more variation in the response than the true model, leading to an underestimation of the residual scale.

Fan et al. (2012) therefore proposes refitted cross-validation (RCV), based on the as-

sumption if a data set is split into multiple chunks, the chance for the same predictor to be spuriously correlated with the response in each chunk is minuscule. Following the idea of cross-validation, variables are selected based on all but one part the data (e.g., using a regularized regression estimator or any other model selection procedure), while the coefficients of the selected predictors are then re-fitted on the left-out part (e.g., using ordinary least squares or a regularized method). To ensure there are enough observations in each part for efficient re-estimation of the coefficients, the data is usually split only into two parts for RCV: $(\mathbf{y}^{(1)}, \mathbf{X}^{(1)})$ and $(\mathbf{y}^{(2)}, \mathbf{X}^{(2)})$. Each of the two parts is used once for model selection, yielding two estimated sets of active predictors, $\hat{\mathcal{A}}^{(1)}$ and $\hat{\mathcal{A}}^{(2)}$. The coefficients are then re-estimated in the other half of the data, restricted to the model selected in the first step. More specifically, the estimate $\hat{\boldsymbol{\theta}}^{(1)}$ is computed using the response vector $\mathbf{y}^{(1)}$ and the subset of the design matrix $\mathbf{X}^{(1)}_{\hat{\mathcal{A}}^{(2)}}$, while $\hat{\boldsymbol{\theta}}^{(2)}$ is computed from $\mathbf{X}^{(2)}_{\hat{\mathcal{A}}^{(1)}}$ and $\mathbf{y}^{(2)}$. The RCV estimate of the residual variance is then the pooled variance estimate

$$s^2_{\text{RCV}} = \frac{\left\|\mathbf{y}^{(1)} - \hat{\mu}^{(1)} - \mathbf{X}^{(1)}_{\hat{\mathcal{A}}^{(2)}}\hat{\boldsymbol{\beta}}^{(1)}\right\|_2^2 + \left\|\mathbf{y}^{(2)} - \hat{\mu}^{(2)} - \mathbf{X}^{(2)}_{\hat{\mathcal{A}}^{(1)}}\hat{\boldsymbol{\beta}}^{(2)}\right\|_2^2}{n - |\hat{\mathcal{A}}^{(1)}| - |\hat{\mathcal{A}}^{(2)}|}.$$

Refitted cross-validation remedies the negative effects of spurious correlation in high dimensions, but the estimation bias introduced by the penalty function is not removed. The effects of data-driven hyper-parameter selection are slightly reduced by decoupling variable selection from coefficient estimation, but they are still noticeable in RCV. Reid et al. (2016) compare RCV for LS-LASSO to other data-splitting methods as well as estimators with folded-concave penalty or other de-biased versions of the LS-LASSO, in a wide range of scenarios. They conclude that estimating the error variance as

$$s^2_{\text{CV}} = \frac{1}{n - |\hat{\mathcal{A}}|}\left\|\mathbf{y} - \hat{\mu} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|_2^2, \tag{5.1}$$

where $\hat{\boldsymbol{\theta}}$ is computed on the full data using a penalization level chosen via standard cross-validation, performs best overall. Theoretical results in Chatterjee and Jafarov (2015) support this conclusion, albeit with a non-adjusted scaling factor $1/n$, which tends to bias the scale estimate downwards. As pointed out in Yu and Bien (2019), the adjustment $\frac{1}{n-|\hat{\mathcal{A}}|}$ is also problematic as it hinges on an accurately recovered model to avoid overestimation of the residuals scale.

Especially when the sparsity of the true signal decreases while the variance explained by the true model remains fixed, Reid et al. (2016) show that RCV and other corrective

measures stop working reliably. With the variance explained by the true model fixed, a less sparse signal also entails decreasing magnitude of each coefficient. Theoretical results for the RCV estimator suggest the magnitude of each truly non-zero coefficient needs to be large enough for the estimator to be consistent and efficient, explaining the results seen by Reid et al. (2016). Surprisingly, Reid et al. (2016) also find that for less sparse models with larger signal strength per coefficient, the RCV estimator is substantially upwards biased in finite samples. Therefore, it appears that correction strategies such as RCV only work well for very sparse problems where the true coefficient values are large enough, which may jeopardize their applicability in practice.

The question is if these empirical results are transferable to PENSE or other robust regularized S-estimators, which bring additional biases. The data-splitting methods can be readily adapted for robust estimation by replacing the regression estimator with, for example, PENSE and the empirical standard deviation by the M-scale of the residuals. The cross-validation based estimator for the residual scale (5.1) may be defined using a PENSE estimate $\tilde{\boldsymbol{\theta}}$ with hyper-parameters selected via cross-validation, and the robust M-scale of the residuals:

$$\hat{\sigma}_{\text{CV}} = \hat{\sigma}_{\text{M}}(\mathbf{y} - \tilde{\mu} - \mathbf{X}\tilde{\boldsymbol{\beta}}).$$

The refitted cross-validation estimator using PENSE for model selection and re-estimation can similarly be defined as

$$\hat{\sigma}_{\text{RCV}} = \sqrt{\frac{1}{2}\left(\hat{\sigma}^2_{\text{M}}(\mathbf{y}^{(1)} - \tilde{\mu}^{(1)} - \mathbf{X}^{(1)}_{\hat{\mathcal{A}}^{(2)}}\tilde{\boldsymbol{\beta}}^{(1)}) + \hat{\sigma}^2_{\text{M}}(\mathbf{y}^{(2)} - \tilde{\mu}^{(2)} - \mathbf{X}^{(2)}_{\hat{\mathcal{A}}^{(1)}}\tilde{\boldsymbol{\beta}}^{(2)})\right)}.$$

Determining an appropriate number of splits for RCV is difficult when using robust estimators. Bisecting the data set leaves enough observations for re-estimating the coefficients but cuts the attainable breakdown point in half. Due to the additional K-fold CV for hyper-parameter selection inside each RCV fold, the maximum attainable breakdown point is $\frac{n(K-1)}{2K}$.

Downward bias of the residual scale can also be exacerbated by overfitting the data. To avoid possible overfitting, the scale of the prediction error could serve as a surrogate estimator for the scale of the residuals:

$$\hat{\sigma}_{\text{PR}} = \hat{\sigma}_{\text{M}}(\mathbf{y} - \hat{\mathbf{y}}^{(\lambda_{\text{S}},\alpha_{\text{S}})})$$

with $\hat{\mathbf{y}}^{(\lambda_{\text{S}},\alpha_{\text{S}})}$ the predicted values in the CV folds as defined in (3.6) and $\lambda_{\text{S}}, \alpha_{\text{S}}$ selected by

the same cross-validation. While the scale of the prediction error may reduce the problem of overfitting, in empirical studies it often overestimates the error scale. An ad hoc way balancing downward bias of $\hat{\sigma}_{\text{CV}}$ and upward bias of $\hat{\sigma}_{\text{PR}}$ is averaging them:

$$\hat{\sigma}_{\text{AVG}} = \sqrt{\frac{\hat{\sigma}^2_{\text{CV}} + \hat{\sigma}^2_{\text{PR}}}{2}}.$$

Despite the lack of theoretical underpinnings, the empirical results presented below indicate that this average estimate performs better than the individual estimates.

It is important to note that here the M-scale estimate

$$\hat{\sigma}_{\text{M}}(\mathbf{r}) = \inf \left\{ s : \frac{1}{n} \sum_{i=1}^{n} \rho(r_i/s) < \delta \right\}$$

is not corrected for the effective degrees of freedom of the estimated model as sometimes done to decrease finite-sample bias of the estimator (e.g., in Maronna 2011). The correction effectively reduces the breakdown point of the M-scale estimate, without adjusting the breakdown point of the robust estimate of regression accordingly. Consider a robust estimate computed with 25% breakdown point, tolerating up to 25% of arbitrarily large residuals. Adjusting the breakdown point of the M-scale estimator, e.g., to 15%, opens the floodgates to some of these possibly extreme residuals affecting the scale estimate and in turn breaking the M-step. As seen before, overestimation of the scale can be even more detrimental to the reliability of the M-estimator than underestimation and should be avoided.

Figure 5.2 summarizes the results of a numerical study of data-splitting methods in conjunction with PENSE. The reported scale estimate is relative to the true scale of the residuals and it is evident that the CV-based estimate, $\hat{\sigma}_{\text{CV}}$, severely underestimates the error scale, especially as more predictors are available. Under contamination, on the other hand, the M-scale of the residuals estimated by PENSE is inflated and shows large variation. At the same time, the estimate based on the prediction error, $\hat{\sigma}_{\text{PR}}$, is badly biased upwards. Under no contamination, under- and overestimation of the CV-based and prediction-based estimates seem to cancel out reasonably well and the average estimate, $\hat{\sigma}_{\text{AVG}}$, performs much better than the individual estimates. In the presence of contaminated observations, however, high variability and upward bias of both $\hat{\sigma}_{\text{CV}}$ and $\hat{\sigma}_{\text{PR}}$ carry over to the average estimate. Refitted cross-validation with PENSE is outperformed by the simple average estimate if no contamination is present but shows slightly better performance under contamination. With RCV, however, the maximum breakdown point is substantially reduced which would

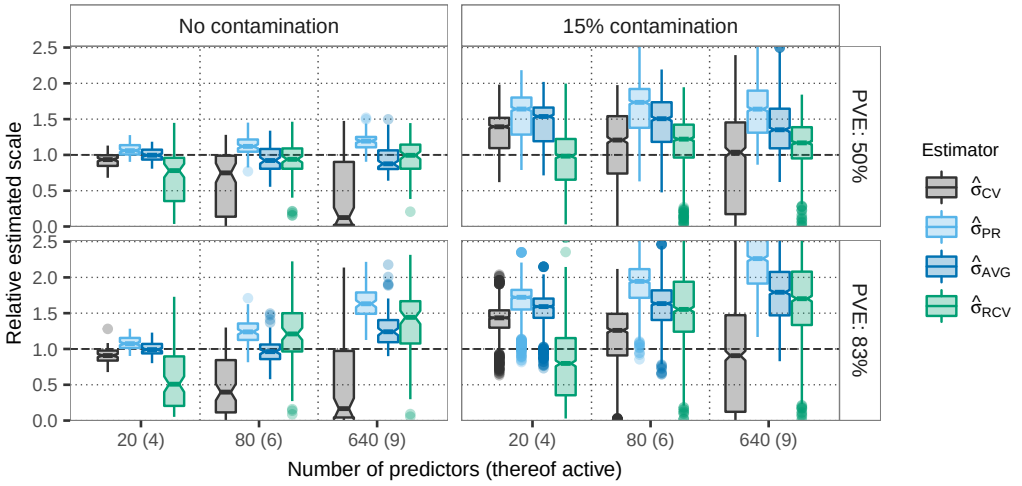


Figure 5.2: Estimated residual scale using PENSE in conjunction with different data-splitting strategies. The reported residual scale estimates are relative to the true scale of the residuals. The $n = 100$ observations are generated according to scheme *VS3-LT** with the true model explaining 50% (top) and 83% (bottom) of the variance. Results under contamination (right) consider scenarios with 10 different vertical outlier positions and moderate leverage, $k_l = 2$.

lead to problems in situations with more than 15% contamination. Concurrent with the findings in Reid et al. (2016), the RCV estimate tends to do worse if the signal strength is larger. The PENSE estimate is perhaps not efficient enough to give reliable estimates for a sample half the size of the original data. In particular due to the stark reduction in the possible breakdown point, RCV is not well suited to be combined with robust estimators. Using adaptive PENSE as the initial high-breakdown estimator instead of PENSE improves results only marginally in these empirical studies and does not warrant the slightly increased computational complexity. The results reported here suggest none of the considered data-splitting strategies works well across all considered scenarios.

5.3 Discussion

The problem of residual scale estimation in moderate- to high-dimensional linear regression models is an actively evolving area. In the context of non-robust regularized estimators, the increased demand for post-selection inference has recently shifted attention to the issue of error variance estimation. Many proposals focus on different data-splitting strategies to get an accurate estimate of the error variance. Others adapt the LS-LASSO for improved residual scale estimation (e.g., Yu and Bien 2019; Sun and Zhang 2012; Belloni et al. 2011), but they explicitly target Normal errors.

For robust estimators, the residual scale estimate has another important role: improving the efficiency of a highly robust but inefficient estimator via a subsequent M-step. This M-step requires an accurate and robust scale estimate to achieve the promised gain in efficiency. As demonstrated empirically in Section 5.1, finite-sample bias in the error scale estimate can render the M-step unreliable. Particularly overestimation of the residual scale exposes the M-estimate to the influence of outliers and hence risks a breakdown under contamination.

Methods for improved scale estimation in the non-robust realm are not transferable to robust regularized estimators due to the effects of possible contamination. Refitted cross-validation and other data-splitting methods for variance estimation suffer from the low efficiency of regularized S-estimators and lead to a severe reduction of the maximum breakdown point. Data-splitting methods for estimating the error variance suffer from the same issues as hyper-parameter search via cross-validation under contamination, discussed in Section 3.5.2.

An interesting direction is presented in Loh (2018) for the L_1 regularized Huber loss, a convex amalgam between the LS- and LAD loss. Up to a fixed threshold, Huber's loss is the square function, which transitions to the absolute value for values greater than the threshold. While not robust towards leverage points in the predictors, it protects against outliers in the response. Choosing the threshold involves the same complications as choosing the cutoff value for the M-step: requiring an estimate of the residual scale. Loh (2018) sidesteps scale estimation and instead proposes to use several candidate values for the scale and adaptively choose a good value based on Lepski's method. The author proves that the resulting estimator performs as well as the estimator obtained by knowing the true error scale. Extensions of this method to regularized M-estimators with redescending ρ_{M} function are of potential interest.

This chapter highlights that estimation of the residual scale in high-dimensions by robust means is very difficult. Methods relying on the accuracy and robustness of a residual scale estimate are susceptible to be severely damaged by contamination. With data-driven hyper-parameter selection, consistency of the scale estimate is not guaranteed, and empirical results suggest the estimates are highly biased. While in pristine settings without any contamination an M-step can indeed improve efficiency and lead to better prediction performance than PENSE or adaptive PENSE, the M-estimator may not be reliable under contamination or heavy-tailed error distributions, overshadowing any potential gain in efficiency. As long as the issue of residual scale estimation in high dimensions is not adequately solved, PENSE and adaptive PENSE are the safer choices over regularized M-estimators.

Chapter 6

Software

As hinted several times in the previous chapters, computing PENSE estimates is a challenging endeavor. For adaptive PENSE, the computational challenges are the same but in general more daunting as adaptive PENSE depends on more hyper-parameters.

To facilitate the application of PENSE (Chapter 3) and adaptive PENSE (Chapter 4), a software package for the language and environment for statistical computing R (R Core Team 2020) is made available at https://cran.r-project.org/package=pense. This chapter details the computational solutions developed for PENSE and adaptive PENSE as available in the `pense` R package. Computation is agnostic to the hyper-parameter ζ, hence it is absorbed by the penalty loadings $\boldsymbol{\omega} = (\omega_1^\zeta, \ldots, \omega_p^\zeta)^\intercal$ and dropped from the notation below. Computation of PENSE is a special case of adaptive PENSE with penalty loadings fixed at $\boldsymbol{\omega} = \mathbf{1}_p$. The following exposition therefore considers only the more general case of adaptive PENSE.

Computing solutions to weighted least-squares adaptive elastic net (LS-adaEN) problems is an essential component for computing adaptive PENSE estimates. As detailed in Chapters 3 and 4, finding a set of initial estimates for PENSE and adaptive PENSE involves a large number of weighted LS-EN and weighted LS-adaEN problems, respectively. Moreover, the adaptive PENSE objective function is equivalent to a weighted LS-adaEN objective function, with weights depending on where the objective function is evaluated. This equivalence is the foundation for computing local minima of the adaptive PENSE objective function. Computation of adaptive PENSE therefore relies heavily on efficient algorithms for solving weighted LS-adaEN problems.

6.1 Algorithms for Weighted LS Adaptive EN

Computational performance of finding local minima of the adaptive PENSE objective function and computing initial estimates depends on the performance of the algorithm for solving weighted LS-adaEN problems of the form

$$\mathscr{O}_{\text{WLS}}(\mu, \boldsymbol{\beta}, \mathbf{W}) = \mathscr{L}_{\text{LS}}\left(\mathbf{W}\mathbf{y}, \mathbf{W}(\mathbf{X}\boldsymbol{\beta} - \mu)\right) + \lambda\Phi_{\text{AN}}(\boldsymbol{\beta}; \boldsymbol{\omega}, \alpha), \tag{6.1}$$

with diagonal weighting matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$. Throughout this section the matrix $\widetilde{\mathbf{W}} = \sqrt{1/\overline{w^2}}\,W$ denotes the normalized weight matrix, where $\overline{w^2} = \frac{1}{n}\sum_{i=1}^{n} W_{ii}^2$ is the average squared weight. Furthermore, the squared matrices $\mathbf{W}^2$ and $\widetilde{\mathbf{W}}^2$ denote the diagonal matrix of squared weights and squared normalized weights, respectively.

Many of the weighted LS-adaEN problems arising during the computation of adaptive PENSE estimates are "close", in the sense that only the weight matrix $\mathbf{W}$ or the set of observations change marginally between subsequent minimizations. While these "proximal" problems are important for adaptive PENSE, computational optimizations for these special use-cases are missing from the literature. Most of the attention in the literature on computing weighted LS-adaEN estimates focuses on computational shortcuts when minimizing the objective function for a decreasing sequence of the penalty parameter (e.g., Friedman et al. 2010; Tibshirani et al. 2012).

In the following, special attention is therefore given to optimizing the weighted LS-adaEN objective function when only the weights or only the data change between subsequent minimizations. Ideally, algorithms for weighted LS-adaEN problems should incur little overhead when changing only weights, data, or the penalty level. The `pense` package implements several algorithms for optimizing the weighted LS-adaEN objective function (6.1), each with its own use-cases, advantages and disadvantages.

6.1.1 Augmented Ridge

The augmented ridge algorithm is specialized for weighted LS-Ridge problems (i.e., $\alpha = 0$ in (6.1)). The weighted LS-Ridge problem can be solved exactly by noting that the weighted LS-adaEN objective function in the case of $\alpha = 0$ and without intercept term simplifies to

$$\frac{1}{2n}\left\|\mathbf{W}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)\right\|_2^2 + \frac{1}{2n}n\lambda\|\boldsymbol{\beta}\|_2^2 = \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}\right\|_2^2 \tag{6.2}$$

where

$$\tilde{\mathbf{y}} = \begin{pmatrix} \mathbf{W}\mathbf{y} \\ \mathbf{0}_p \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{W}\mathbf{X} \\ \sqrt{n\lambda}\mathbf{I}_{p\times p} \end{pmatrix}.$$

Due to the equivalence in 6.2, the closed-form solution for the Ridge estimate $\hat{\boldsymbol{\beta}}$ is

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\intercal\mathbf{W}^2\mathbf{X} + n\lambda\mathbf{I}_{p\times p}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}^2\mathbf{y}.$$

An intercept term can be accommodated by making the predictor matrix orthogonal to the centered response. More specifically, the weighted and centered response is $\mathbf{y}^* = \mathbf{W}\mathbf{y} - \frac{1}{n}\mathbf{1}_n^\intercal\mathbf{W}\mathbf{y}$. Similarly, the orthogonalized predictor matrix $\mathbf{X}^*$ is given by

$$\mathbf{X}^* = \tilde{\mathbf{X}} - \frac{1}{n}\mathbf{W}\mathbf{1}_{n\times n}\mathbf{W}\tilde{\mathbf{X}} \tag{6.3}$$

where $\tilde{\mathbf{X}} = \mathbf{W}(\mathbf{X} - \mathbf{1}_p\bar{\mathbf{x}})$ is the centered and weighted predictor matrix and $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}^\intercal\mathbf{1}_n$ is the mean vector of all predictors. The slope parameter and intercept are then computed by

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \left(\mathbf{X}^{*\intercal}\mathbf{X}^* + n\lambda\mathbf{I}_{p\times p}\right)^{-1}\mathbf{X}^{*\intercal}\mathbf{y}^* \\ \hat{\mu} &= \frac{1}{n}\mathbf{1}_n^\intercal\mathbf{W}^2\left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right). \end{aligned} \tag{6.4}$$

Computing the optimum for any penalty level incurs $O(np+p^3)$ floating-point operations (flops) to solve the system of p linear equations in (6.4). Changing the data or weights requires recomputing the orthogonalized predictor matrix $\mathbf{X}^*$ and its Gram matrix $\mathbf{X}^{*\intercal}\mathbf{X}^*$. These changes therefore incur an additional computational complexity of $O(n^2p + np^2)$ flops. Solving the linear equations in (6.4) can be a computational bottleneck for very large p. However, if the $p \times p$ matrix fits into memory and $\lambda > 0$, the augmented Ridge algorithm is highly competitive as the solution can be computed to high precision in a single step without potential convergence issues. This stability argument often outweighs limited scalability as computing local minima of the adaptive PENSE objective function involves a large number of weighted LS-adaEN problems and convergence issues in a single weighted LS-adaEN problem lead to more serious convergence issues down the road.

6.1.2 Augmented LARS

The Least Angle Regression (Efron et al. 2004) algorithm (LARS) can be used to compute solutions of the LS-LASSO objective function exactly. Starting from the empty model, i.e., all coefficients equal 0, the LARS algorithm translates a LS-LASSO problem into a sequence

of ordinary least-squares (OLS) problems, one for each penalty level where covariates "enter" or "leave" the model. For a fixed penalty level λ, the LARS algorithm solves $K \geq 0$ OLS problems at $\tilde{\lambda}_0 > \tilde{\lambda}_1 > \ldots > \tilde{\lambda}_K$, where $\tilde{\lambda}_{K-1} < \lambda \leq \tilde{\lambda}_K$. The LS-LASSO at penalty level λ can then be recovered exactly by linear interpolation between the coefficients computed at $\tilde{\lambda}_{K-1}$ and $\tilde{\lambda}_K$:

$$\hat{\boldsymbol{\beta}}(\lambda) = \frac{\tilde{\lambda}_K - \lambda}{\tilde{\lambda}_K - \tilde{\lambda}_{K-1}} \hat{\boldsymbol{\beta}}(\tilde{\lambda}_{K-1}) + \frac{\lambda - \tilde{\lambda}_{K-1}}{\tilde{\lambda}_K - \tilde{\lambda}_{K-1}} \hat{\boldsymbol{\beta}}(\tilde{\lambda}_K).$$

Penalty loadings $\boldsymbol{\omega}$ are incorporated into the LARS algorithm by scaling the predictor matrix with the inverse penalty loadings $\mathbf{X}\boldsymbol{\Omega}^{-1}$, where $\boldsymbol{\Omega}^{-1} = \text{diag}(1/\omega_1, 1/\omega_2, \ldots, 1/\omega_p)$. The elastic net penalty can be accommodated by changing the penalty level for the LARS algorithm to $\alpha\lambda$ and using equivalence (6.2) to handle the L_2 penalization with $\sqrt{n\lambda}\mathbf{I}_{p\times p}$ replaced by matrix $\sqrt{n\frac{1-\alpha}{2}\lambda}\boldsymbol{\Omega}^{-1}$ The LARS algorithm therefore solves the weighted LS-adaEN problem by computing the LS-LASSO solution on the weighted, centered response and orthogonalized predictors given in (6.3), and $\mathbf{X}$ replaced by $\mathbf{X}\boldsymbol{\Omega}^{-1}$.

At every step k, $k = 0, \ldots, K$, of the augmented LARS algorithm a system of linear equations must be solved. However, the sequence of the OLS problems allows for solving these systems of linear equations more efficiently by sequentially updating a "running" Cholesky decomposition (Efron et al. 2004; Watkins 2002). Consider the symmetric $p \times p$ matrix $\mathbf{A} = \mathbf{X}^{*\intercal}\mathbf{X}^* + \sqrt{n\frac{1-\alpha}{2}\lambda}\boldsymbol{\Omega}^{-1}$. In the following, $\mathbf{A}^{(k)}$ denotes the symmetric matrix comprising only the rows and columns of $\mathbf{A}$ for predictors included in the model at the k-th step. Instead of calculating $\mathbf{A}^{(k)}$ for every k, the augmented LARS algorithm only needs the (upper-triangular) Cholesky decomposition $\mathbf{U}^{(k)}$ of $\mathbf{A}^{(k)}$, $\mathbf{A}^{(k)} = \mathbf{U}^{(k)\intercal}\mathbf{U}^{(k)}$. This Cholesky decomposition can be computed efficiently from the Cholesky decomposition at the previous step, $\mathbf{U}^{(k-1)}$. Consider predictor j is added in the k-th step. The updated Cholesky decomposition is given by

$$\mathbf{U}^{(k)} = \begin{pmatrix} \mathbf{U}^{(k-1)} & {\mathbf{U}^{(k-1)}}^{-1}\mathbf{v} \\ \mathbf{0}^\intercal & A_{jj} - \mathbf{v}^\intercal\mathbf{v} \end{pmatrix}, \quad \mathbf{v} = \left(A_{jj'}\right)_{j' \in \mathscr{A}^{(k-1)}} \tag{6.5}$$

where $\mathscr{A}^{(k-1)}$ is the set of active predictors in the previous step. The system ${\mathbf{U}^{(k-1)}}^{-1}\mathbf{v}$ can be solved efficiently using back substitution because $\mathbf{U}^{(k-1)}$ is an upper-triangular matrix (Watkins 2002). This requires only $O(\tilde{p}^2)$ operations, where $\tilde{p} \leq p$ is the dimension of $\mathbf{U}^{(k-1)}$. Performing updates in this way leads to a different order of the predictors in the

Cholesky decomposition than in $\mathbf{X}$. Therefore, it is necessary to keep track of the order the predictors are added to reconstruct the original order of the coefficients. The performance gain, however, outweighs the overhead of rearranging the coefficients only once.

Dropping a predictor is also a simple update of the Cholesky decomposition. Consider predictor j is dropped in the k-th step, and $\mathbf{v} = (\mathbf{v}_1^\intercal, v_2)^\intercal$ corresponds to the upper-diagonal elements of the column in $\mathbf{U}^{(k-1)}$ corresponding to the dropped predictor,

$$\mathbf{U}^{(k-1)} = \begin{pmatrix} \mathbf{U}_{11}^{(k-1)} & \mathbf{v}_1 & \mathbf{U}_{13}^{(k-1)} \\ 0 & v_2 & U_{23}^{(k-1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_{33}^{(k-1)} \end{pmatrix}.$$

The updated Cholesky decomposition $\mathbf{U}^{(k)}$ is then given by

$$\mathbf{U}^{(k)} = \begin{pmatrix} \mathbf{U}_{11}^{(k-1)} & \mathbf{U}_{13}^{(k-1)} \\ \mathbf{0} & \mathbf{U}_{33}^{(k)} \end{pmatrix}$$

where $\mathbf{U}_{33}^{(k)}$ is the Cholesky decomposition of a rank-one update $\mathbf{U}_{33}^{(k-1)\intercal}\mathbf{U}_{33}^{(k-1)} + \mathbf{v}_1\mathbf{v}_1^\intercal$, which can be computed efficiently (Gill et al. 1974).

Updating the running Cholesky decomposition involves growing and shrinking of the decomposition at every single step. Conventionally, the $\tilde{p}^2$ elements of the decomposition $\mathbf{U} \in \mathbb{R}^{\tilde{p}\times\tilde{p}}$ are stored in a contiguous array (Anderson et al. 1999):

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1\tilde{p}} \\ u_{21} & u_{22} & \cdots & u_{2\tilde{p}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\tilde{p}1} & u_{\tilde{p}2} & \cdots & u_{\tilde{p}\tilde{p}} \end{pmatrix} \xrightarrow{\text{stored as}} [\underbrace{u_{11}, u_{21}, \ldots, u_{\tilde{p}1}}_{\text{column 1}}, \underbrace{u_{12}, u_{22}, \ldots, u_{\tilde{p}2}}_{\text{column 2}}, \ldots, \underbrace{u_{1\tilde{p}}, u_{2\tilde{p}}, \ldots, u_{\tilde{p}\tilde{p}}}_{\text{column } \tilde{p}}].$$

This storage schema is not ideal for the running Cholesky decomposition for two reasons. First, the decomposition is an upper-triangular matrix with all entries below the diagonal being 0 and never referenced. Therefore, the conventional storage scheme requires almost twice as much memory as necessary. Secondly, appending or removing a column and row to/from a conventionally stored matrix requires moving almost every element in memory, which is an expensive operation. Considering that any row appended to the Cholesky decomposition contains only 0's, except for the diagonal entry, this is a superfluous and prodigal operation.

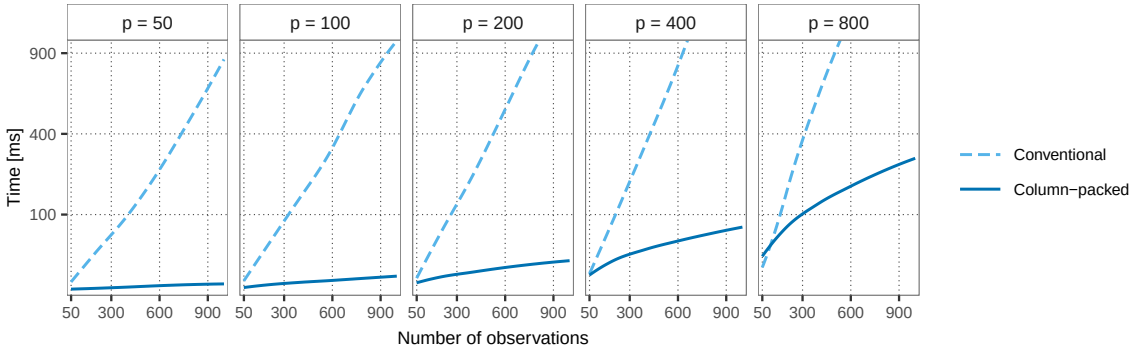


Figure 6.1: Comparison of computation time for the weighted LS-adaEN minimizer using the augmented LARS algorithm with the Cholesky decomposition stored in *conventional* scheme (dashed light-blue line) or *column-packed* scheme (solid blue line). The vertical axis is on the square-root-scale. Timings are taken for simulated data sets (one per (n, p) combination) and averaged over 100 runs on a system with Intel® Xeon® E5-1650 v2 @ 3.50GHz processors.

To improve performance of the running Cholesky decomposition used for the augmented LARS algorithm, the implementation in the `pense` package stores the decomposition a in column-packed scheme (Anderson et al. 1999). Only the $(\tilde{p}^2 + \tilde{p})/2$ non-zero elements of the upper-triangular Cholesky decomposition $\mathbf{U} \in \mathbb{R}^{\tilde{p} \times \tilde{p}}$ are stored in memory as

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1\tilde{p}} \\ 0 & u_{22} & \cdots & u_{2\tilde{p}} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{\tilde{p}\tilde{p}} \end{pmatrix} \xrightarrow{\text{stored as}} \Big[\underbrace{u_{11}}_{\text{column 1}}, \underbrace{u_{12}, u_{22}}_{\text{column 2}}, \ldots, \underbrace{u_{1\tilde{p}}, u_{2\tilde{p}}, \ldots, u_{\tilde{p}\tilde{p}}}_{\text{column } \tilde{p}} \Big].$$

Appending a row and column to the matrix $\mathbf{U}$ only requires appending $\tilde{p} + 1$ elements in memory, without moving any of the other elements. Removing a row and column from matrix $\mathbf{U}$ still requires moving elements in memory, but it is less expensive than for conventional storage as only non-zero elements must be moved. Considering that appending is a much more frequent operation than removing for the running Cholesky decomposition (Efron et al. 2004), the performance gains of using column-packed storage are substantial. This is evident in Figure 6.1, where the computation times for two implementations of the augmented LARS algorithm are compared: an implementation using the conventional storage scheme for the Cholesky decomposition (denoted by "conventional" in the graph) and an improved implementation using column-packed storage for the Cholesky decomposition (denoted by "column-packed"). For most problem sizes, the "column-packed" storage scheme leads to substantial improvements in computational speed.

Augmented LARS solves the optimization of the weighted LS-adaEN objective function exactly using a sequence of OLS problems and is therefore numerically very stable. Unless $\alpha = 1$, changing the penalty parameters requires recomputing the entire sequence of OLS problems. Each update of the running Cholesky decomposition requires $O(\tilde{p}^2)$ flops, where $\tilde{p} \leq p$ is the number of active predictors in the step. Therefore, computational complexity for solving the sequence of K OLS problems is $O(Kp^2)$, where K is typically $\lesssim \max(n, p)$ unless predictors are highly correlated. Furthermore, if the penalty level is large and hence the solution has a small number of non-zero coefficients, the augmented LARS algorithm involves only a few low-dimensional OLS problems and is computationally very efficient. As for augmented Ridge, updating the weights or data requires recomputing the weighted, orthogonal predictor matrix $\mathbf{X}^*$ adding $O(n^2p + np^2)$ flops. Quadratic computational complexity can also be seen in Figure 6.1. On the square-root scaling of the vertical axis in these plots the computation time increases linearly with the number of observations n, for any p.

Closed form solutions for the intermediate OLS problems avoid convergence issues for augmented LARS. Accurate results, high stability and computational efficiency for sparse solutions (i.e., large penalty levels) are clear advantages of the augmented LARS algorithm. A main drawback, however, is the need to store a $p \times p$ matrix and, for small penalty levels, $O(p^2)$ flops per step. Furthermore, the algorithm cannot leverage solutions to "proximal" problems (e.g., after a small change to the penalty level) to speed up computation, a key advantage of iterative algorithms.

6.1.3 Alternating Direction Method of Multipliers (ADMM)

The Alternating Direction Method of Multipliers (ADMM) algorithm leverages the fact that the objective function of weighted LS-adaEN (6.1) is compound of the convex weighted LS loss and the non-smooth (but convex) adaptive EN penalty. For ADMM, the minimization problem is written in consensus form (Deng and Yin 2016)

$$\begin{aligned} \underset{\mu,\boldsymbol{\beta}}{\arg\min}\, \mathscr{O}_{\text{WLS}}(\mu, \boldsymbol{\beta}) = \quad & \underset{\boldsymbol{\theta}\in\mathbb{R}^{p+1}, \hat{\mathbf{y}}\in\mathbb{R}^n}{\arg\min} \quad f(\hat{\mathbf{y}}) + g(\boldsymbol{\theta}) \\ & \text{subject to} \quad \hat{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta} = 0 \end{aligned} \tag{6.6}$$

with $f(\hat{\mathbf{y}}) = \frac{1}{2}\|\widetilde{\mathbf{W}}(\mathbf{y} - \hat{\mathbf{y}})\|_2^2$ the (scaled) weighted LS loss function, $\tilde{\mathbf{X}} = (\mathbf{1}_n, \mathbf{X})$ the predictor matrix with a column of 1's for the intercept term, and $g(\boldsymbol{\theta})$ the scaled adaptive EN penalty

function

$$g(\boldsymbol{\theta}) = g((\mu, \boldsymbol{\beta}^\intercal)^\intercal) = \frac{n}{\bar{w}} \lambda \Phi_{\text{AN}}(\boldsymbol{\beta}; \boldsymbol{\omega}, \alpha).$$

The consensus form splits the optimization problem for $\hat{\mathbf{y}}$ and $\boldsymbol{\theta}$ in two independent parts and one equality constraint. The constrained optimization problem in (6.6) can be cast into an unconstrained augmented Lagrangian problem

$$L_\tau(\boldsymbol{\theta}, \hat{\mathbf{y}}, \mathbf{z}) = f(\hat{\mathbf{y}}) + g(\boldsymbol{\theta}) + \mathbf{z}^\intercal(\hat{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta}) + \frac{\tau}{2}\|\hat{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta}\|_2^2$$

with step size $\tau > 0$ and dual variable $\mathbf{z} \in \mathbb{R}^n$ for the consensus constraint (Bertsekas 1982; Deng and Yin 2016).

In the augmented Lagrangian formulation of the minimization problem, parameters $\hat{\mathbf{y}}$ and $\boldsymbol{\theta}$ are separable up to a quadratic term. The augmented Lagrangian problem is solved iteratively by

$$\boldsymbol{\theta}^{(k+1)} = \arg\min_{\boldsymbol{\theta}} L_\tau(\hat{\mathbf{y}}^{(k)}, \boldsymbol{\theta}, \mathbf{z}^{(k)}) \tag{6.7}$$

$$\hat{\mathbf{y}}^{(k+1)} = \arg\min_{\hat{\mathbf{y}}} L_\tau(\hat{\mathbf{y}}, \boldsymbol{\theta}^{(k+1)}, \mathbf{z}^{(k)}) \tag{6.8}$$

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \tau\left(\hat{\mathbf{y}}^{(k+1)} - \tilde{\mathbf{X}}\boldsymbol{\theta}^{(k+1)}\right) \tag{6.9}$$

where $k > 0$ is the iteration counter.

The challenge computing the first step (6.7) in the ADMM iterations stems from of the product $\tilde{\mathbf{X}}\boldsymbol{\theta}$ in the quadratic penalty term. To simplify the step, it can be approximated by linearizing the quadratic term $\frac{\tau}{2}\|\hat{\mathbf{y}}^{(k)} - \tilde{\mathbf{X}}\boldsymbol{\theta}\|_2^2$ by a first-degree Taylor expansion around $\boldsymbol{\theta}^{(k)}$:

$$\begin{aligned}\frac{\tau}{2}\|\hat{\mathbf{y}}^{(k)} - \tilde{\mathbf{X}}\boldsymbol{\theta}\|_2^2 &\propto \tau(\hat{\mathbf{y}}^{(k)} - \tilde{\mathbf{X}}\boldsymbol{\theta}^{(k)})^\intercal\tilde{\mathbf{X}}\boldsymbol{\theta} + \tau\|\tilde{\mathbf{X}}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)})\|_2^2 \\ &< \tau\left((\hat{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\theta}^{(k)})^\intercal\tilde{\mathbf{X}}\boldsymbol{\theta} + \frac{1}{2\tau'}\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\|_2^2\right)\end{aligned}$$

with $0 < \tau' < 1/\|\tilde{\mathbf{X}}\|^2$ (He and Yuan 2015). Instead of (6.7), this "linearized" ADMM

solves the minimization problem

$$\begin{aligned}
\boldsymbol{\theta}^{(k+1)} &= \arg\min_{\boldsymbol{\theta}} L_\tau(\hat{\mathbf{y}}^{(k)}, \boldsymbol{\theta}, \mathbf{z}^{(k)}) \\
&= \arg\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta}) + \tau \left(\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)} \right)^\intercal \tilde{\mathbf{X}}^\intercal \left(\tilde{\mathbf{X}} \boldsymbol{\theta}^{(k)} - \hat{\mathbf{y}}^{(k)} + \frac{1}{\tau} \mathbf{z}^{(k)} \right) + \frac{1}{2\tau'} \|\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\|_2^2 \right) \\
&= \arg\min_{\mu, \boldsymbol{\beta}} \; g((0, \boldsymbol{\beta}^\intercal)^\intercal) \\
&\qquad + \tau \left(\boldsymbol{\beta}^\intercal \mathbf{X}^\intercal \left(\mathbf{X} \boldsymbol{\beta}^{(k)} - \hat{\mathbf{y}}^{(k)} + \frac{1}{\tau} \mathbf{z}^{(k)} \right) n \mu^{(k)} \boldsymbol{\beta}^\intercal \bar{\mathbf{x}} + \frac{1}{2\tau'} \|\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}\|_2^2 \right) \\
&\qquad + \mu\tau \left(n\mu^{(k)} + n\bar{\mathbf{x}}^\intercal \boldsymbol{\beta}^{(k)} + \sum_{i=1}^{n} \left(\frac{1}{\tau} z_i^{(k)} - \hat{y}_i^{(k)} \right) \right) + \frac{\tau}{2\tau'} \left(\mu - \mu^{(k)} \right)^2
\end{aligned}$$

where $\bar{\mathbf{x}} \in \mathbb{R}^p$ is the vector of column means of the predictor matrix $\mathbf{X}$. This minimization problem can be solved separately for the intercept and slope. The updated intercept using the linear approximation is

$$\mu^{(k+1)} = \mu^{(k)} - \tau \left(n\mu^{(k)} + n\bar{\mathbf{x}}^\intercal \boldsymbol{\beta}^{(k)} + \sum_{i=1}^{n} \left(\frac{1}{\tau} z_i^{(k)} - \hat{y}_i^{(k)} \right) \right). \tag{6.10}$$

The updated slope can be represented by the proximal operator of the adaptive EN penalty:

$$\boldsymbol{\beta}^{(k+1)} = \mathbf{prox}_{\frac{\tau' n\lambda}{\tau \bar{w}} \Phi_{\mathrm{AN}}} \left(\boldsymbol{\beta}^{(k)} - \tau \mathbf{X}^\intercal \left(\mathbf{X} \boldsymbol{\beta}^{(k)} - \hat{\mathbf{y}}^{(k)} + \frac{1}{\tau} \mathbf{z}^{(k)} \right) - n\tau \mu^{(k)} \bar{\mathbf{x}} \right). \tag{6.11}$$

Following Parikh and Boyd (2014), the proximal operator $\mathbf{prox}_{\eta f} \colon \mathbb{R}^q \to \mathbb{R}^q$ of a closed proper convex function $f \colon \mathbb{R}^q \to \mathbb{R}$, scaled by positive scalar $\eta \in \mathbb{R}_+$, is defined as

$$\mathbf{prox}_{\eta f}(\mathbf{u}) = \arg\min_{\mathbf{v} \in \mathbb{R}^q} \left\{ f(\mathbf{v}) + \frac{1}{2\eta} \|\mathbf{u} - \mathbf{v}\|_2^2 \right\}.$$

The proximal operator of the EN penalty is thus the scaled, coordinate-wise, soft-thresholding operator (Parikh and Boyd 2014):

$$\mathbf{prox}_{\eta \Phi_{\mathrm{AN}}}(\mathbf{u}) = \left(\frac{\operatorname{sgn}(u_j) \max(0, |u_j| - \eta\alpha\omega_j)}{1 + \eta(1 - \alpha)} \right)_{j=1}^{p}. \tag{6.12}$$

Once the first step in the ADMM iterations is computed, the second step (6.8), can be

easily solved by

$$\hat{\mathbf{y}}^{(k+1)} = \arg\min_{\hat{\mathbf{y}}} L_\tau(\hat{\mathbf{y}}, \boldsymbol{\theta}^{(k+1)}, \mathbf{z}^{(k)}) = \left(\mathbf{I}_{n\times n} + \frac{1}{\tau}\widetilde{\mathbf{W}}^2\right)^{-1}\left(\tilde{\mathbf{X}}\boldsymbol{\theta}^{(k+1)} + \frac{1}{\tau}\left(\widetilde{\mathbf{W}}^2\mathbf{y} + \mathbf{z}^{(k)}\right)\right)$$

and involves only the inverse of a diagonal matrix. The final step (6.9) is a simple vector update and does not incur substantial computations.

A single iteration for linearized ADMM can be computed very efficiently, requiring only $O(pn)$ flops. The convergence rate and hence the number of iterations of the linearized ADMM algorithm depends on the rank of the predictor matrix $\mathbf{X}$ as well as the elastic net parameter α. Deng and Yin (2016) show that if either $\mathbf{X}$ has full column rank or $\alpha < 1$ (i.e., the EN penalty is strongly convex), $\boldsymbol{\theta}^{(k)}$ converges "Q-linearly" to a global minimum $\boldsymbol{\theta}^*$, meaning there exists a $c \in (0, 1)$ such that

$$\frac{\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|_2}{\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|_2} \leq c.$$

In the case where $\alpha = 1$ (i.e., adaptive LASSO) and $\mathbf{X}$ does not have full column rank, the convergence rate of linearized ADMM is only sub-linear (Davis and Yin 2017), in the sense that the value of the objective function converges sub-linearly to the value of the objective function at a global minimum,

$$(f(\hat{\mathbf{y}}^{(k)}) + g(\boldsymbol{\theta}^k)) - (f(\tilde{\mathbf{X}}\boldsymbol{\theta}^*) + g(\boldsymbol{\theta}^*)) = O(1/k).$$

Theoretically, linearized ADMM converges for any choice of the step size parameter τ. The actual speed of convergence of linearized ADMM, however, depends heavily on the value chosen for τ. If τ is too small or too large, the algorithm may not converge within a reasonable number of iterations or even diverge due to numerical instability. The convergence rates in Deng and Yin (2016) can be used to determine an "optimal" step size if $\tilde{\mathbf{X}}$ is of full column rank or $\alpha < 1$. In the case where both conditions are satisfied, the optimal step size is the product of the minimum and maximum weights, $\tau = \min_i \widetilde{W}_{ii} \times \max_i \widetilde{W}_{ii}$. In case neither condition is satisfied, the step size is more difficult to tune, and no theoretical guidance is available.

Steps 6.10, 6.11, 6.8, and 6.9 are iterated until the gap between iterations is sufficiently small, i.e.,

$$\|\hat{\mathbf{y}}^{(k+1)} - \hat{\mathbf{y}}^{(k)}\|_2^2 + \|\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\|_2^2 < \epsilon$$

for a small convergence threshold $\epsilon > 0$, or until the algorithm exceeds the prespecified maximum number of iterations.

Overall, linearized ADMM can be very efficient, but a change to the data requires computing the "linearization" step size τ', incurring an additional $O(p^2 n)$ flops. The main advantage of linearized ADMM is that a single iteration is very efficient and that it can leverage solutions to "proximal" problems. However, convergence can be very slow if the step size is not chosen properly.

6.1.4 Dual Augmented Lagrangian (DAL)

The DAL algorithm as proposed in Tomioka et al. (2011) is an iterative algorithm which can be adapted to computing the weighted LS-adaEN estimate. Using the same functions f and g as defined for the ADMM algorithm (6.6), DAL uses Fenchel's duality theorem (Rockafellar 1970, Theorem 31.1) to cast the weighted LS-adaEN objective

$$\underset{\mu,\boldsymbol{\beta}}{\arg\min}\, \mathscr{O}_{\mathrm{WLS}}(\mu, \boldsymbol{\beta}) = \underset{\mu\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, f(\mathbf{X}\boldsymbol{\beta} + \mu\mathbf{1}_n) + g(\boldsymbol{\beta})$$

into its corresponding dual form

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^n,\mathbf{v}\in\mathbb{R}^p}{\arg\max} \quad & -f^*(-\boldsymbol{\alpha}) - g^*(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{v} = \mathbf{X}^\intercal\boldsymbol{\alpha} \quad \text{and} \quad \mathbf{1}_n^\intercal\boldsymbol{\alpha} = 0 \end{aligned}$$

where the second equality constraint encodes the intercept. The functions f^* and g^* are the convex conjugates of f and g, respectively, and defined as

$$f^*(\mathbf{v}) = \sup_{\mathbf{u}\in\mathbb{R}^n} \left(\mathbf{v}^\intercal\mathbf{u} - f(\mathbf{u})\right), \qquad g^*(\mathbf{v}) = \sup_{\mathbf{u}\in\mathbb{R}^p} (\mathbf{v}^\intercal\mathbf{u} - g(\mathbf{u})).$$

As the name suggests, Dual Augmented Lagrangian iteratively minimizes the augmented Lagrangian of this dual problem, given by

$$L_\tau(\boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\beta}) = -f^*(\boldsymbol{\alpha}) - g^*(\mathbf{v}) + \boldsymbol{\beta}^\intercal(\mathbf{v} - \mathbf{X}^\intercal\boldsymbol{\alpha} - \mathbf{1}_n^\intercal\boldsymbol{\alpha}) - \frac{\tau}{2}\|\mathbf{v} - \mathbf{X}^\intercal\boldsymbol{\alpha}\|_2^2.$$

In Fenchel's dual formulation the Lagrangian multiplier, $\boldsymbol{\beta}$, corresponds to the primal solution to the weighted LS-adaEN problem (for the slope), and the intercept can be easily recovered by $\mu = \tau\mathbf{1}_n^\intercal\boldsymbol{\alpha}$.

Algorithm 3 Dual augmented Lagrangian algorithm for the weighted LS-adaEN problem.

Input: Initial step size multiplier $\eta > 0$, initial solution $\boldsymbol{\beta}^{(0)}$, $\mu^{(0)}$.

1: $\tau_1^{(0)} = \tau_2^{(0)} = \eta\overline{w^2}/(n\lambda)$
2: $\boldsymbol{\alpha}^{(0)} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(0)}$
3: **repeat**
4: $\quad \boldsymbol{\beta}^{(k+1)} = \mathbf{prox}_{\frac{n}{\tau_1^{(k)}\vec{w}}\lambda\Phi_{\mathrm{AN}}}\left(\boldsymbol{\beta}^{(k)} + \tau_1^{(k)}\mathbf{X}\boldsymbol{\alpha}^{(k)}\right)$
5: $\quad \mu^{(k+1)} = \mu^{(k)} + \tau_2^{(k)}\mathbf{1}_n^\intercal\boldsymbol{\alpha}^{(k)}$
6: $\quad \tau_1^{(k+1)} = 2\tau_1^{(k)}$
7: $\quad$ **if** $k > 1$ and $|\mathbf{1}_n^\intercal\boldsymbol{\alpha}^{(k+1)}| > \epsilon$ and $|\mathbf{1}_n^\intercal\boldsymbol{\alpha}^{(k+1)}| > |\mathbf{1}_n^\intercal\boldsymbol{\alpha}^{(k)}|/2$ **then**
8: $\quad\quad \tau_2^{(k+1)} = 10\tau_2^{(k)}$
9: $\quad$ **else**
10: $\quad\quad \tau_2^{(k+1)} = 2\tau_2^{(k)}$
11: $\quad$ **end if**
12: $\quad \boldsymbol{\alpha}^{(k+1)} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \varphi_{k+1}(\boldsymbol{\alpha})$, where

$$\varphi_{k+1}(\boldsymbol{\alpha}) = f^*(-\boldsymbol{\alpha}) + \frac{1}{2\tau_1^{(k+1)}}\left\|\mathbf{prox}_{\frac{n}{\tau_1^{(k+1)}\vec{w}}\lambda\Phi_{\mathrm{AN}}}\left(\boldsymbol{\beta}^{(k+1)} + \tau_1^{(k+1)}\mathbf{X}\boldsymbol{\alpha}\right)\right\|_2^2 + \frac{1}{2\tau_2^{(k+1)}}\left(\mu^{(k+1)} + \tau_2^{(k+1)}\mathbf{1}_n^\intercal\boldsymbol{\alpha}\right)^2$$

13: $\quad k = k + 1$
14: **until** $\mathrm{RDG}^{(k)} < \epsilon$ (as defined in (6.13))

Tomioka et al. (2011) propose to solve this dual augmented Lagrangian problem by the iterative procedure given in Algorithm 3. In the first step on lines 4 and 5, $\boldsymbol{\beta}^{(k+1)}$ and $\mu^{(k+1)}$ are updated from the previous solution using the dual vector $\boldsymbol{\alpha}^{(k)}$. The slope $\boldsymbol{\beta}^{(k+1)}$ is updated through the proximal operator of the adaptive EN penalty as given in (6.12) and together with the update to the intercept term can be done in $O(pn)$ flops. The second step updates the step sizes τ_1 and τ_2 for the slope and intercept, respectively. The last step, updating the dual vector $\boldsymbol{\alpha}^{(k+1)}$, is more involved; the strongly convex function φ_{k+1} can only be minimized approximately using numerical methods. The DAL algorithm implemented in the `pense` package uses Newton's method with backtracking line search (Boyd et al. 2004, pp. 464ff) for computing an approximate solution $\boldsymbol{\alpha}^{(k+1)}$. Newton's method for minimizing φ_{k+1} requires inverting the $n \times n$ Hessian of φ_{k+1} and hence a total of $O(n^3 + n^2p)$ flops. This can be somewhat improved by noting that the Hessian of φ_{k+1} changes only marginally between iterations and the inversion can be accelerated by using the previous inverse as a pre-conditioner in the conjugate gradient method (Gentle 2007,

Algorithm 6.2).

To get exponential convergence of the DAL algorithm the step size needs to increase at every iteration. Furthermore, to alleviate convergence issues due to the unpenalized intercept, Algorithm 3 implements the suggestion in Tomioka et al. (2011) to use separate step sizes for the slope coefficients ($\tau_1^{(k)}$) and the intercept coefficient ($\tau_2^{(k)}$). If the intercept coefficient does not change substantially between iterations, the step size for the intercept is increased aggressively to speed up convergence.

The DAL algorithm is stopped when the relative duality gap, $\text{RDG}^{(k)}$ is less than the prescribed numerical tolerance $\epsilon > 0$. The relative duality gap is defined as

$$\text{RDG}^{(k)} = \frac{f(\mathbf{X}\boldsymbol{\beta}^{(k)} + \mu^{(k)}\mathbf{1}_n) + g(\boldsymbol{\beta}^{(k)}) - f^*(-\tilde{\boldsymbol{\alpha}}^{(k)}) - g^*(\mathbf{X}^\intercal\tilde{\boldsymbol{\alpha}}^{(k)})}{f(\mathbf{X}\boldsymbol{\beta}^{(k)} + \mu^{(k)}\mathbf{1}_n) + g(\boldsymbol{\beta}^{(k)})} \tag{6.13}$$

with candidate dual vector $\tilde{\boldsymbol{\alpha}}^{(k)} = \boldsymbol{\alpha}^{(k)} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\intercal\boldsymbol{\alpha}^{(k)}$.

Tomioka et al. (2011) establish strong convergence results for the DAL algorithm, even when solving for $\boldsymbol{\alpha}^{(k+1)}$ only approximately. The DAL algorithm converges super-linearly to a global optimum, $\boldsymbol{\theta}^*$, of the weighted LS-adaEN objective, i.e.,

$$\frac{\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|_2}{\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|_2} \leq \frac{1}{\sqrt{1 + 2c\tau_1^{(k)}}},$$

for some constant $c > 0$. It can be seen that convergence is faster the larger the initial step size $\tau_1^{(0)}$, but a larger step size makes the optimization of φ_{k+1} more difficult as the strong convexity constant of φ_{k+1} is inversely related to $\tau_1^{(k+1)}$. The default setting in the `pense` package is to double the step size in each iteration, as shown in Algorithm 3. The initial step size is derived from the level of penalization and the scale of the loss function multiplied by parameter $\eta > 0$, using a conservative multiplier of $\eta = 0.01$ by default. Compared to ADMM, DAL is designed to converge in much fewer iterations, but each iteration carries a substantially higher computational burden. The advantages of DAL are threefold: (i) DAL performs noticeably better for (severely) ill-conditioned problems than other iterative algorithms (Tomioka et al. 2011), (ii) DAL is well suited when the number of predictors p is much larger than the number of observations n and (iii) sparsity in the primal solution vector $\boldsymbol{\beta}$ can be harnessed to substantially reduce the memory footprint and computational complexity.

The faster convergence of DAL is clearly visible Figure 6.2 for two simulated data

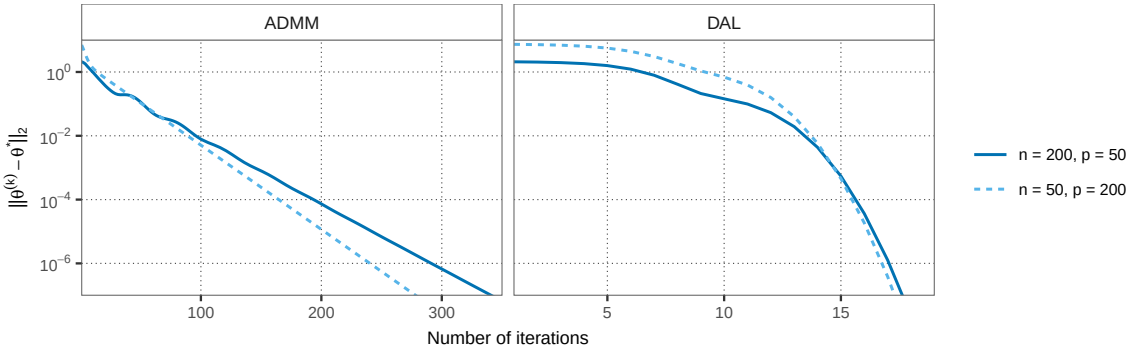


Figure 6.2: Distance between the true global minimum, $\boldsymbol{\theta}^*$, and the solution in the k-th iteration, $\boldsymbol{\theta}^{(k)}$ versus iteration counter k for linearized ADMM and DAL for weighted LS-adaEN on two data sets simulated according to scheme *MS1-MH(-2, 8)*. Observation weights, w_i $(i = 1, \ldots, n)$ are random draws from a uniform distribution on $[0, 4]$ and the penalty loadings ω_j $(j = 1, \ldots, p)$ are from a uniform distribution on $[0, 1]$.

sets with randomly generated observation weights and penalty loadings. The exact global minima for these two data sets are computed using the augmented LARS algorithm up to floating-point precision. The hyper-parameters of the adaptive EN penalty are fixed at $\alpha = 0.5$ and $\lambda = \bar{\lambda}_{\text{WLS}}/2$, where $\bar{\lambda}_{\text{WLS}}$ is the smallest penalty level such that $\boldsymbol{\beta} = \mathbf{0}_p$ minimizes the weighted LS-adaEN objective function. As summarized in Table 6.1, linearized ADMM exhibits linear convergence for $\alpha < 1$, which is supported by the linear trend under logarithmic scaling of the distance between the iterates $\boldsymbol{\theta}^{(k)}$ and the true global minimum $\boldsymbol{\theta}^*$. DAL, on the other hand, converges super-linearly and requires far fewer iterations than ADMM to get within a distance of 10^{-6} of the true global minimum. In terms of computational speed, however, DAL only outperforms ADMM if the number of observations is small and the number of predictors is very large.

Table 6.1 summarizes computational complexity of the algorithms implemented in the `pense` package. They are optimized to perform well in the use-cases required for computing adaptive PENSE estimates. Particular attention is devoted to reducing the overhead incurred by small changes to the data, for example changing weights between successive minimizations. These three algorithms for weighted LS-adaEN cover a wide range of problem sizes and ensure computing adaptive PENSE estimates is feasible in applications with large and demanding data sets.

	Augmented LARS	Linearized ADMM	DAL
Complexity	$O(n^2p + np^2 + Kp^2)$	$O(Kpn)$	$O(K(n^3 + n^2p))$
Data-change overhead	–	$O(p^2n)$	–
# of iterations, K	$\lesssim \max(n, p)$	$O(e^{-k})$ or $O(1/k)$	$o(e^{-k})$

Table 6.1: Comparison of computational complexity of algorithms to minimize the weighted LS-adaEN objective function (6.1) measured in floating-point operations. For augmented LARS, the number of steps required K is usually the number of non-zero coefficient values in the result, but in the presence of highly correlated predictors the number of iterations may be slightly larger. Linearized ADMM converges linearly, in $O(e^{-k})$ iterations, if the penalty function is strictly convex (i.e., $\alpha < 1$) or if $\mathbf{X}^\intercal\mathbf{X}$ is positive definite.

6.2 Initial Estimates

The non-convex objective function of adaptive PENSE bears the need for an elaborate scheme to find good starting points. These starting points, or "initial estimates", are a crucial component of computing regularized S-estimates. Numerical methods for finding local minima of the non-convex objective function 4.1 converge to different local stationary points depending on the chosen starting point. Different strategies are explored in Section 3.2, while the most reliable strategy for regularized S-estimates is the EN-PY procedure detailed in algorithms 1 and 2.

The computational burden of EN-PY is substantial due to the computation of leave-one-out (LOO) residuals required to compute the sensitivity matrix $\mathbf{R}$ (line 2 in Algorithm 2) and because LS-adaEN estimates need to be computed for each potentially clean subset of the data (line 7 in Algorithm 1). As detailed in Section 3.2.4, it is difficult to match the level of penalization desired for adaptive PENSE with an appropriate level of penalization for the EN-PY procedure. Therefore, EN-PY initial estimates are usually computed for a fixed α but a set of Q penalty levels $\mathscr{Q}_I$.

In case of multiple penalty levels, line 4 of Algorithm 1 can be improved upon in the first iteration ($\iota = 0$) because the index set $\mathscr{I}^{(0)}$ is the same for all penalty levels. Iterative algorithms for optimizing the LS-adaEN objective function, such as ADMM and DAL discussed in Section 6.1, at penalty level λ_q, $1 < q < Q$, converge faster if the minimum of the LS-adaEN objective function at penalty level λ_{q-1} is leveraged. A similar improvement in the first iteration can be implemented for computing LOO LS-adaEN estimates needed for the sensitivity matrix $\mathbf{R}$. For subsequent iterations such optimizations are not possible because the index set $\mathscr{I}^{(\iota)}$ is most likely different for different penalty levels. However, the iterations can be done in parallel for different penalty levels, leveraging multiple cores with

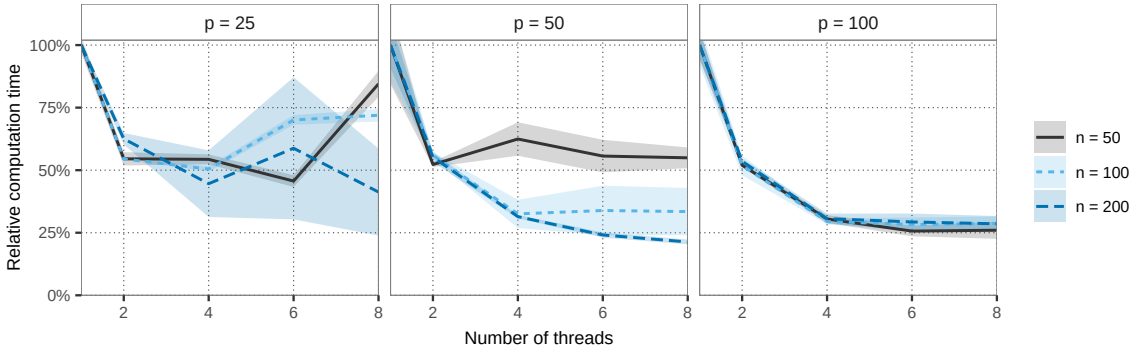


Figure 6.3: Comparison of the average time to compute the EN-PY initial estimates using 1 to 8 threads. Computation time is relative to the average computation time required using 1 thread. Timings are taken for data simulated according to scheme *MS1-MH(-2, 8)* and averaged over 100 runs on a system with Intel® Xeon® E3-12XX @ 2.70GHz processors (each CPU comprises 4 cores). Augmented LARS is used to compute LS-adaEN solutions and penalty parameters are fixed at $\alpha_{\text{AS}} = 0.5$, $\boldsymbol{\omega} = \mathbf{1}_p$. The set $\mathscr{Q}_I = \{5 \times 10^{-4}\widetilde{\lambda}_{\text{AS}}, \ldots, \widetilde{\lambda}_{\text{AS}}\}$ contains 12 penalty levels, equally spaced on the logarithmic scale, with $\widetilde{\lambda}_{\text{AS}}$ given in (6.21).

negligible overhead because these computations are completely independent.

Figure 6.3 shows the speed gains of using $1 - 8$ CPU cores simultaneously via threads for computing the EN-PY initial estimates over a grid of 12 penalization levels, starting at the smallest penalty level such that $\mathbf{0}_p$ is a local optimum, as given in (6.21). For each combination of n and p, a single data set is randomly generated according to data generation scheme *MS1-MH(-2, 8)* and computation is replicated 100 times. The system has 9 processors with 4 cores each, i.e., sharing data between 4 threads incurs little overhead, while moving beyond 4 threads involves increased memory management. This is also visible in Figure 6.3, where performance does not improve noticeably when using more than 4 threads, even for large problems. For all problem sizes, two threads can reduce computation time almost by half, while for small problems the overhead of more threads can devour the gains of parallelizing. In general, the more challenging the problem, the more gains from multithreading. If possible, using as many threads as cores per processor leads to fastest computation without degrading performance.

Iterations of the EN-PY procedure must be done sequentially, but some steps within a single iteration allow for efficient parallelization to multiple cores. Computing the LS-adaEN estimates on the potentially clean subsets (line 7 in Algorithm 1) can be performed in parallel without the need to share data between cores. Similarly, the LOO estimates used for the sensitivity matrix $\mathbf{R}$ can be computed simultaneously on multiple cores.

In case of the Ridge penalty ($\alpha = 0$), EN-PY initial estimates can be computed much

faster by exploiting the linearity of the LS-Ridge estimator. Instead of computing LOO residuals manually, the elements of the sensitivity matrix $\mathbf{R}$ can be computed efficiently by $R_{ij} = \mathbf{y}^\intercal \mathbf{H}_{i\cdot} - H_{ij} e_j / (1 - H_{jj})$ where

$$\mathbf{H} = \mathbf{X} \left(\mathbf{X}^\intercal \mathbf{X} + (n-1)\lambda I \right)^{-1} \mathbf{X}^\intercal \quad \text{and} \quad \mathbf{e} = \mathbf{y} - \mathbf{H}\mathbf{y}.$$

The closed-form solution for the sensitivity matrix considerably improves computational speed for EN-PY in case of the Ridge penalty. However, the Ridge penalty does not lead to any coefficient value being exactly 0. Therefore, all eigenvalues of $\mathbf{R}^\intercal \mathbf{R}$ are non-zero ($Q = \tilde{n}$, the number of observation in the EN-PY iteration), leading to a large number of potentially clean subsets and hence the need to compute many LS-adaEN estimates in line 7 of Algorithm 1.

The EN-PY procedure given in Algorithm 1 returns only the estimates from the last iteration. The risk of missing potentially good initial estimates can be reduced by tweaking the algorithm to additionally retain all estimates "close" to the best initial estimate, $\hat{\boldsymbol{\theta}}^{(\iota)}$, from the final iteration (in terms of their M-scale of the residuals). The EN-PY procedure implemented in the `pense` package retains estimates from all previous iterations which have less than twice the M-scale of the residuals from the best initial estimate. The threshold can be changed to retain more or less estimates from previous iterations. Retaining estimates from previous iterations increases the computational burden but boosts the chances of finding global optima.

The main computational challenge for EN-PY is solving a large number of LS-adaEN subproblems. Furthermore, numerical instability or convergence issues of algorithms are difficult to correct automatically but can have a detrimental effect on EN-PY. It is therefore important to employ efficient and stable numerical algorithms, chosen according to the dimension of the sample. Section 6.1 details the algorithms available in the `pense` package. Computation can be accelerated by leveraging "proximity" of LS-adaEN problems arising in the EN-PY procedure. When computing LOO estimates, for example, the estimates are unlikely to differ drastically from each other. Therefore, the computational burden can be substantially decreased by leveraging the LOO estimate $\hat{\boldsymbol{\theta}}_{(-(i-1))}$ when computing the LOO estimate $\hat{\boldsymbol{\theta}}_{(-i)}$, $i = 2, \ldots, \tilde{n}$ in line 2 of Algorithm 2.

The algorithm for solving the LS-adaEN subproblems needs to be chosen in accordance with the dimensions of the problem. Figure 6.4 shows computation time for EN-PY initial estimates using different algorithms to solve the LS-adaEN subproblems for several com-

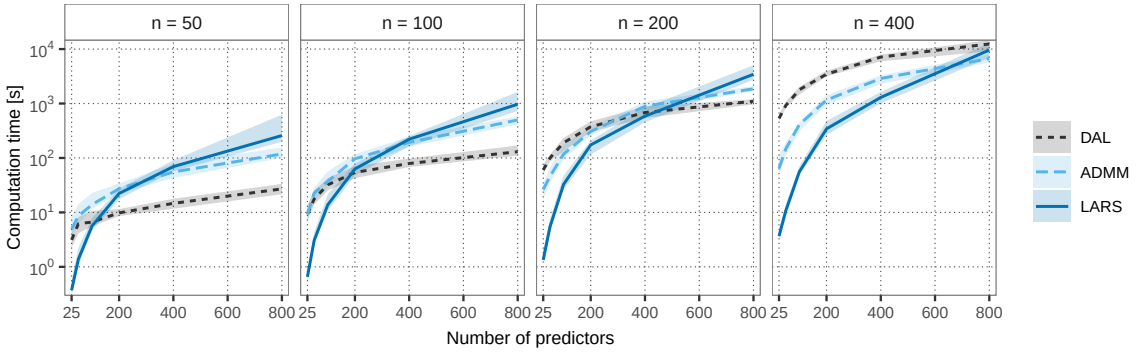


Figure 6.4: Comparison of the median time (on log-scale) for computing EN-PY initial estimates using different algorithms to the solve LS-adaEN subproblems. The shaded areas depict the inter-quartile range over 50 replications on a system running on Intel® Xeon® CPU E3-12XX @ 2.70GHz processors. Data is simulated according to scheme *MS1-MH(-2, 8)* with varying number of observations (n) and predictors (p). Penalty parameters are fixed at $\alpha_{\text{AS}} = 0.5$, $\boldsymbol{\omega} = \mathbf{1}_p$, and the set $\mathscr{Q}_I(\alpha) = \{5 \times 10^{-4}\widetilde{\lambda}_{\text{AS}}, \ldots, \widetilde{\lambda}_{\text{AS}}\}$ contains 12 penalty levels, equally spaced on the logarithmic scale, with $\widetilde{\lambda}_{\text{AS}}$ given in (6.21).

binations of the number of observations, n, and number of predictors p. As suggested by the computational complexity of the different algorithms in Table 6.1, the DAL algorithm outperforms others if the number of observations is reasonably small but the number of parameters is large. The DAL algorithm leverages proximal solutions particularly well, often requiring only one or two iterations when computing LOO estimates, making it particularly well suited for the EN-PY procedure as long as n is not too large. The LARS algorithm, on the other hand, does not benefit from proximal solutions but giving its efficient implementation it is usually the fastest option if the number of predictors is small to moderate. Computational complexity of linearized ADMM is linear in both n and p, but because changing the data incurs additional $O(p^2 n)$ flops, ADMM is recommended for EN-PY only if both n and p are large.

For each λ in the set of penalty levels, $\mathscr{Q}_I$, the EN-PY procedure yields a set of initial estimates $\mathscr{T}(\lambda)$. Due to the difficulty of matching the penalty level between the EN-PY procedure and adaptive PENSE, the implementation in the `pense` package combines all initial estimates into one large set of initial estimates $\mathscr{T} = \bigcup_{\lambda \in \mathscr{Q}_I} \mathscr{T}(\lambda)$. Each of these initial estimates is subsequently used to find local minima of the adaptive PENSE objective function.

6.3 Computing Local Minima

Once a set of reliable starting points, $\mathscr{T}$, is obtained the task is to locate local minima of the adaptive PENSE objective function (4.1) close to these starting points. The adaptive PENSE objective function is not continuously differentiable everywhere, making gradient-based methods or Newton's method unusable (Parikh and Boyd 2014). Subgradient-based methods are a generalization of gradient-based methods for non-smooth functions (Shor 1985). Subgradient-based methods are conceptually simple, but convergence to local stationary points is generally slow and not ascertained for the non-convex adaptive PENSE objective function (Bagirov et al. 2013). While some adaptations of subgradient-based methods improve convergence for non-convex problems (e.g., Bagirov et al. 2013), they are in practice unstable for large-scale problems. For adaptive PENSE, the most stable and efficient numerical algorithms are based on the Minimization by Majorization (MM) principle.

MM algorithms are a broad class of algorithms with many applications. Lange (2016) provides an extensive overview of the theory and applications of MM algorithms. The general idea of MM algorithms is very versatile yet simple. For adaptive PENSE, for instance, the goal is to find a local minimum of the objective function $\mathscr{O}_{\text{AS}}(\boldsymbol{\theta})$ over $\boldsymbol{\theta} \in \mathbb{R}^{p+1}$, starting from an initial guess $\boldsymbol{\theta}^{(0)}$. Key to MM algorithms is finding a "surrogate" function with majorizes the true objective function at anchor point $\boldsymbol{\theta}^*$. A function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ is said to majorize the objective function $\mathscr{O}_{\text{AS}}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$ if

$$g(\boldsymbol{\theta}^*|\boldsymbol{\theta}^*) = \mathscr{O}_{\text{AS}}(\boldsymbol{\theta}^*) \quad \text{and} \quad g(\boldsymbol{\theta}|\boldsymbol{\theta}^*) \geq \mathscr{O}_{\text{AS}}(\boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta} \in \boldsymbol{\theta} \in \mathbb{R}^{p+1}. \tag{6.14}$$

In other words the majorizing surrogate function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ equals the true objective function at $\boldsymbol{\theta}^*$ and is greater than the true objective function everywhere else. An MM algorithm sequentially minimizes surrogate functions until a fixed point of the true objective function is reached. Starting from the initial guess $\boldsymbol{\theta}^{(0)}$, the sequence of steps is given by

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{p+1}}{\arg\min}\, g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}) \tag{6.15}$$

for $k = 0, 1, \ldots$ until $d\left(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}\right) < \epsilon$ where $d : \mathbb{R}^{p+1} \times \mathbb{R}^{p+1} \to [0, \infty)$ is a distance metric and $\epsilon > 0$ a numerical tolerance level. Iterations of MM algorithms are guaranteed

to produce a sequence of estimates with non-increasing value of the objective function:

$$\mathscr{O}_{\text{AS}}(\boldsymbol{\theta}^{(k+1)}) \leq g(\boldsymbol{\theta}^{(k+1)}|\boldsymbol{\theta}^{(k)}) \leq g(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k)}) = \mathscr{O}_{\text{AS}}(\boldsymbol{\theta}^{(k)}). \tag{6.16}$$

The first inequality and last equality are due to g being a majorizing function and the middle inequality holds because $\boldsymbol{\theta}^{(k+1)}$ minimizes $g(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$. For a suitably chosen surrogate function, the iterates (6.15) converge at least sub-linearly to a stationary point of the true objective function close to the initial guess $\boldsymbol{\theta}^{(0)}$ (Lange 2016). This stationary point does not have to be a local minimum, but because the adaptive PENSE objective is optimized for a multitude of starting points, saddle points and local maxima are very likely screened out at the end.

The idea is that a difficult problem (i.e., finding local minima of the true objective function) is replaced by a sequence of simpler problems (i.e., finding minima of surrogate functions). This implies that the surrogate function $g(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ must be reasonably simple and easy to minimize for MM algorithms to be of use. For adaptive PENSE, it suffices to find a surrogate function for the S-loss, as the adaptive EN penalty is already convex. From 6.14 it is evident that combining a majorizer of the S-loss with the adaptive EN penalty majorizes the entire adaptive PENSE objective function.

The local representation of the objective function as a weighted adaptive LS-EN problem, introduced first in Section 3.1, proves important for deriving a surrogate function of the adaptive PENSE objective function. Let $\tilde{\mathbf{X}} = (\mathbf{1}_n, \mathbf{X}) \in \mathbb{R}^{n\times(p+1)}$ be the predictor matrix augmented by a column of 1's for the intercept term. For any anchor point $\boldsymbol{\theta}^* \in \mathbb{R}^{p+1}$, consider the local surrogate function

$$\begin{aligned} g_{\text{S}}(\boldsymbol{\theta}|\boldsymbol{\theta}^*) &= \frac{1}{2n}\left\|\mathbf{W}_{\boldsymbol{\theta}^*}\left(\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\theta}\right)\right\|_2^2 + \lambda_{\text{AS}}\Phi_{\text{AN}}(\boldsymbol{\beta};\boldsymbol{\omega},\alpha_{\text{AS}}) \\ &= \mathscr{O}_{\text{WLS}}(\boldsymbol{\theta}, \mathbf{W}_{\boldsymbol{\theta}^*}). \end{aligned} \tag{6.17}$$

with diagonal weight matrix $\mathbf{W}_{\boldsymbol{\theta}^*} \in \mathbb{R}^{n\times n}$ having diagonal elements

$$w_i = \sqrt{\frac{\rho'(\tilde{r}_i)/\tilde{r}_i}{\frac{1}{n}\sum_{k=1}^n \rho'(\tilde{r}_k)\tilde{r}_k}} \quad \text{where } \tilde{r}_i = \frac{y_i - \tilde{\mathbf{x}}_i^\intercal\boldsymbol{\theta}^*}{\hat{\sigma}_{\text{M}}(\boldsymbol{\theta}^*)}, \quad i = 1,\ldots,n.$$

It is easy to verify that $g_{\text{S}}(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ coincides with the adaptive PENSE objective function at $\boldsymbol{\theta}^*$, but this surrogate function is not ascertained to majorize the objective function everywhere. Following Fan et al. (2018), it is not necessary for the surrogate to majorize

the true objective function everywhere for the MM algorithm to produce a converging sequence of iterates. The sequence converges as long as the surrogate majorizes the true objective function locally, i.e., satisfies the local property

$$\mathscr{O}_{\text{AS}}(\boldsymbol{\theta}^{(k+1)}) \leq g(\boldsymbol{\theta}^{(k+1)}|\boldsymbol{\theta}^{(k)}). \tag{6.18}$$

The MM algorithm implemented in the `pense` package utilizes the weighted LS-adaEN surrogate function as defined in (6.17) despite the lack of proof that the local property (6.18) holds. If at any iteration property (6.18) is violated, the iteration can be repeated using a shifted and scaled weighted LS-adaEN surrogate function until the local property is satisfied. In practice an instance where the local property is violated by the surrogate (6.17) has yet to emerge, suggesting that the surrogate does indeed satisfy the local property. Local minima of the adaptive PENSE can therefore be computed efficiently by sequentially solving weighted LS-adaEN problems.

Numerical tolerance for solving LS-adaEN problems

These weighted LS-adaEN problems are simpler than the non-convex adaptive PENSE objective function, but they are not solvable exactly either. Many numerical algorithms to solve weighted LS-adaEN problems do so up to a prescribed numerical tolerance. From the non-increasing sequence in 6.16 it can be seen that the surrogate functions do not have to be minimized exactly as long as iterates $\boldsymbol{\theta}^{(k+1)}$ reduce (or at least not increase) the surrogate objective function.

This observation opens avenues for improving performance of MM algorithms. Considering a desired numerical tolerance for local optima of ϵ as defined below (6.15), only the last MM iteration must solve the surrogate problem with numerical tolerance less than ϵ, preceding iterations can solve the surrogate problems with less accuracy. The idea is in the same spirit as the continuous analogue of the MM principle discussed in Lange (2016, p. 110), without requiring a strictly convex or smooth surrogate function. To improve numerical stability, the surrogate problem must be solved with higher accuracy than ϵ in the final iterations. The implementation in the `pense` package solves the surrogate problems in the final iteration with a more stringent numerical tolerance of $\tilde{\epsilon} = \epsilon/10$. Using less accurate iterations generally increases the number of MM iterations required to find local optima, but at the same times decreases the computational burden of minimizing the surrogate function. The actual speed improvement depends on the strategy to choose the accuracy

for MM iterates and on the computational complexity of initializing the algorithm for the surrogate problem with "reweighted" data and savings from weaker demands on accuracy.

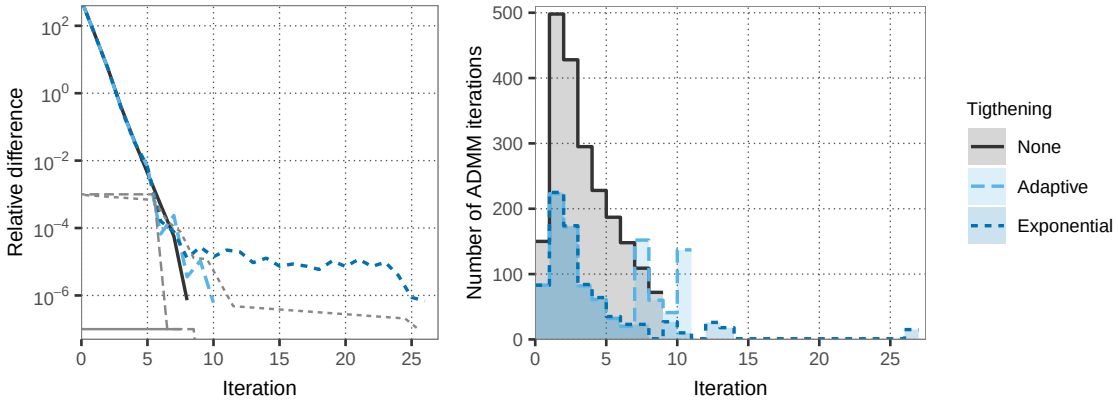
The `pense` package implements two "tightening" strategies to reduce computation time: exponential and adaptive. Exponential tightening sets the initial numerical tolerance level to $\epsilon^{(0)} = \sqrt{\tilde{\epsilon}}$. If the surrogate objective decreases in the k-th MM iteration, in other words, $g(\boldsymbol{\theta}^{(k+1)}|\boldsymbol{\theta}^{(k)}) < g(\boldsymbol{\theta}^{(k)}|\boldsymbol{\theta}^{(k)})$, the numerical tolerance is adjusted to

$$\epsilon^{(k+1)} = \max\left(\epsilon, \min\left(d(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}), \epsilon^{(k)}\tilde{\epsilon}^{2/K}\right)\right),$$

where K is the maximum number of MM iterations. If the surrogate objective function is not decreased, the iteration is repeated with a smaller numerical tolerance, i.e., $\epsilon^{(k)} = \epsilon^{(k)}\tilde{\epsilon}^{1/10}$.

Adaptive tightening, on the other hand, decreases the numerical tolerance for the subproblems only if the parameter does not change meaningfully. As for exponential tightening, the initial numerical tolerance level is $\epsilon^{(0)} = \sqrt{\tilde{\epsilon}}$. The "aggressiveness" of adaptive tightening, and what is considered a meaningful change in the parameters, is controlled through the maximum number of adjustments S, with a default value of $S = 1$. If the surrogate objective decreases in the k-th MM iteration but the parameter values do not change substantially, the numerical tolerance remains constant, i.e., $\epsilon^{(k+1)} = \epsilon^{(k)}$. Adaptive tightening takes action if the surrogate objective decreases in the k-th MM iteration and the change in parameter values, $d(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) < \epsilon^{(k)}$, adjusting the tolerance to $\epsilon^{(k+1)} = \epsilon^{(k)}\tilde{\epsilon}^{1/S}$. In case the surrogate objective function does not decrease, the iteration is repeated with a tighter numerical tolerance $\epsilon^{(k)} = \epsilon^{(k)}\tilde{\epsilon}^{1/(2S)}$.

The effect of these different tightening strategies is shown in Figure 6.5 for a single simulated data set with desired convergence tolerance $\epsilon = 10^{-6}$. The plot on the left shows the relative difference in the value of the adaptive PENSE objective function between consecutive iterations as well as the convergence tolerance for the surrogate problem, $\epsilon^{(k)}$. Without tightening strategy (solid black line), the convergence tolerance for the surrogate problem remains fixed at $\tilde{\epsilon} = 10^{-7}$, in which case the MM algorithm converges after 7 iterations. With adaptive tightening (dashed light-blue line), the number of MM iterations increases to 10, and for exponential tightening (dotted blue line) 26 MM iterations are required. While the tightening schemes lead to more MM iterations, the total number of iterations performed by ADMM are 1851, 617, and 583 for no tightening, adaptive tightening, and exponential tightening, respectively. The plot on the right highlights that tightening strategies reduce the number of ADMM iterations especially for the first few MM iterations. At



(a) Relative difference to coefficients from previous iteration.

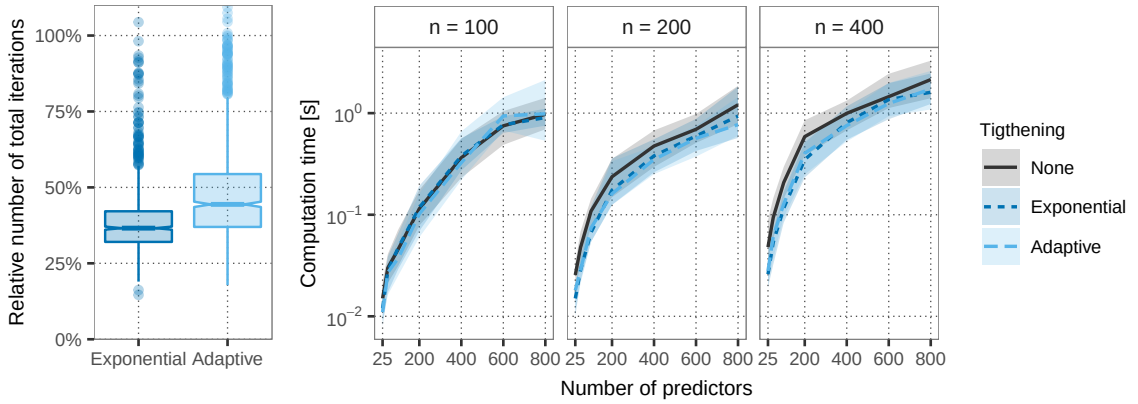
(b) Number of ADMM iterations.

Figure 6.5: Convergence path of different tightening strategies for the MM algorithm for adaptive PENSE. The weighted LS-adaEN solutions are computed using linearized ADMM. Data is generated according to scheme *MS1-MH(-2, 8)* with 100 observations and 400 predictors. The gray lines in plot (a) depict the numerical tolerance to solve the surrogate problems for the different tightening strategies at each iteration. Penalty parameters are fixed at $\alpha_{\text{AS}} = 0.5$, $\boldsymbol{\omega} = \mathbf{1}_p$, and $\lambda_{\text{AS}} = \widetilde{\lambda}_{\text{AS}}/2$, with $\widetilde{\lambda}_{\text{AS}}$ given in (6.21). The MM algorithm is started at $\mathbf{0}_{p+1}$ and the convergence tolerance is $\epsilon = 10^{-6}$.

these initial iterations, the MM iterates change considerably and it is not necessary to solve the surrogate function precisely. Once the MM iterations approach the local minimum of the adaptive PENSE objective function, however, more precise solutions are necessary to avoid "zigzagging" around the local minimum.

Figure 6.5(b) also shows the numerical tolerance level at each MM-iteration, $\epsilon^{(k)}$, visualizing how tightening strategies work. As described above, adaptive tightening reduces the numerical tolerance of the surrogate problem once the relative change between iterates is smaller than $\epsilon^{(k)}$. After one adjustment, adaptive tightening uses the maximum accuracy of $\tilde{\epsilon} = 10^{-7}$. From the right plot it can further be seen that as soon as the numerical tolerance is lowered, ADMM requires substantially more iterations. With exponential tightening, on the other hand, the numerical tolerance changes more gradually and ADMM needs in general less iterations to converge in the individual MM iterations. At the very end the numerical tolerance is reduced to the desired accuracy of $\tilde{\epsilon} = 10^{-7}$, leading to slightly more ADMM iterations.

The smoother adjustment of the numerical tolerance for exponential tightening leads in general to a lower number of ADMM iterations. This trend is also visible in Figure 6.6(a), where the total number of ADMM iterations required per adaptive PENSE minimization (relative to the number of ADMM iterations required if no tightening strategy is used) are



(a) Number of ADMM iterations.

(b) Computation time.

Figure 6.6: Performance of the MM algorithm (using the linearized ADMM algorithm to minimize the surrogate functions) for computing local minima of the adaptive PENSE objective function, using different tightening strategies. *Figure (a):* Number of ADMM iterations required to compute a local minimum of the adaptive PENSE objective function for different tightening strategies, relative to the number of ADMM iterations required with no tightening strategy. *Figure (b):* Median runtime of the MM algorithm with different tightening strategies to compute a local minimum of the adaptive PENSE objective function. The vertical axis is on the log-scale. The shaded area around the median depicts the inter-quartile range from 50 replications measured on a system with Intel® Xeon® E3-12XX processors clocked at 2.70GHz. Data is generated according to scheme *MS1-MH(-2, 8)* and penalty parameters are fixed at $\alpha_{\text{AS}} = 0.5$, $\boldsymbol{\omega} = \mathbf{1}_p$, and $\lambda_{\text{AS}} = \widetilde{\lambda}_{\text{AS}}/2$, with $\widetilde{\lambda}_{\text{AS}}$ given in (6.21). The MM algorithm is started at $\mathbf{0}_{p+1}$ and the convergence tolerance is set to 10^{-6}.

compared for exponential and adaptive tightening. Both tightening strategies lead to a substantial decrease in the total number of ADMM iterations required, with exponential tightening leading to a slightly greater reduction. This translates to decreased computation time as evident in Figure 6.6(b), where the time required to compute a minimum of the adaptive PENSE objective function is shown for different problem sizes. Albeit the overall reduction in computation time is not as pronounced as the reduction in ADMM iterations, tightening saves computing resources especially for large problems.

Tightening works well with linearized ADMM, but less so with DAL. Adaptive tightening slightly reduces the number of DAL iterations required, but exponential tightening increases the number of DAL iterations substantially, almost tripling the number of DAL iterations in the numerical experiments for Figure 6.6. The reason for this inflation of DAL iterations is that the convergence criterion employed by DAL (the relative duality gap) is not linearly related to the relative change in the coefficient values as used by the MM algorithm to determine convergence. Furthermore, if the weights change, the inner minimization carried out for DAL (step 12 of Algorithm 3) cannot re-use the Hessian from the previous iteration,

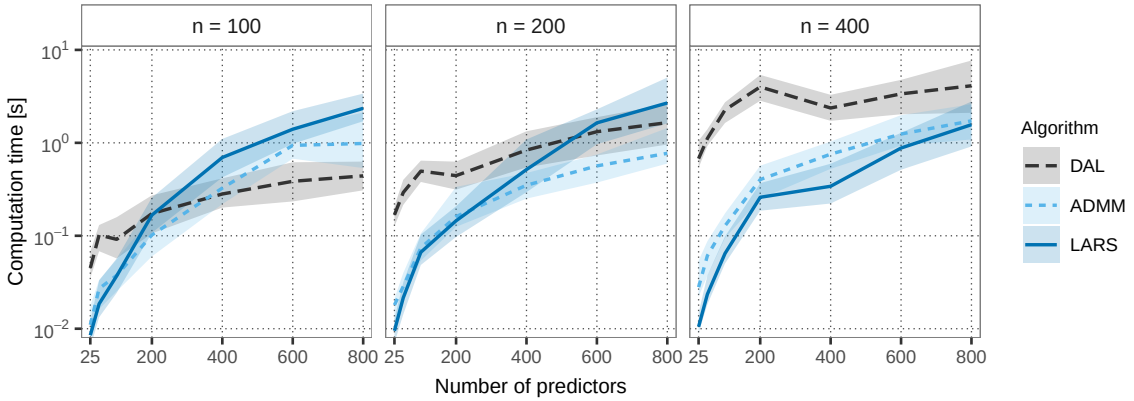


Figure 6.7: Median time for computing local minima of the adaptive PENSE objective function (4.1) by the MM algorithm using different algorithms to solve the weighted LS-adaEN subproblems. The MM algorithm uses adaptive tightening for ADMM, while no tightening is used for DAL and LARS algorithms. The vertical axis is on the log-scale and the shaded area around the median depicts the inter-quartile range from 50 replications measured on a system with Intel® Xeon® E3-12XX processors clocked at 2.70GHz. Data is generated according to scheme *MS1-MH(-2, 8)* and penalty parameters are fixed at $\alpha_{\text{AS}} = 0.5$, $\boldsymbol{\omega} = \mathbf{1}_p$, and $\lambda_{\text{AS}} = \widetilde{\lambda}_{\text{AS}}/2$, with $\widetilde{\lambda}_{\text{AS}}$ given in (6.21). The MM algorithm is started at $\mathbf{0}_{p+1}$ with convergence tolerance set to 10^{-6}.

leading to overhead in the computations which cannot be compensated by a moderate reduction in the number of DAL iterations through tightening.

Performance of the MM algorithm with each of the three algorithms for weighted LS-adaEN described in Section 6.1 is shown in Figure 6.7. It is noticeable that the augmented LARS algorithm outperforms the other algorithms for small p or large n. As expected, the DAL algorithm is competitive for a small number of observations and when the number of predictors is large. However, as already noted above, changing weights causes the DAL algorithm to recompute the Hessian required for the inner minimization from scratch. Therefore, DAL is better suited for use in the EN-PY procedure where changes to the data are more gradual than in the MM algorithm, except for scenarios with many predictors and few observations. Linearized ADMM, on the other hand, strikes a balance between augmented LARS and DAL and is suggested for situations where both n and p are moderate to large. An important property of the augmented LARS algorithm which is not visible in these plots is its accuracy. While augmented LARS is often outperformed by iterative algorithms, iterative algorithms are more prone to convergence issues, leading in turn to convergence problems for the MM algorithm.

The MM algorithm developed for adaptive PENSE delivers reliable and scalable performance. Allowing the use of any algorithm for solving weighted LS-adaEN subproblems, the MM algorithm is adaptable to many problems. Tightening strategies further reduce com-

putational complexity of solving a large number of subproblems with iterative algorithms.

These optimizations become even more important when the MM algorithm is run numerous times. The algorithm described in this chapter locates a local minimum for fixed hyper-parameters and a single starting point. In practice, a large set of different starting points needs to be explored to increase chances of finding a global optimum of the objective function. Furthermore, good values for the hyper-parameters are unknown in advance and need to be selected in a data-driven fashion, involving multitudinous minimizations. The solutions developed for the MM algorithm and weighted LS-adaEN algorithms are crucial to make large-scale explorations possible, but there is room for even more aggressive optimizations.

6.4 Computing Adaptive PENSE for Many Hyper-Parameters

As detailed in previous chapters, good values for the hyper-parameters of PENSE and adaptive PENSE are in practice unknown and need to be selected based on the available data. Sections 3.5 and 4.1.1 outline the benefits and shortcomings of using K-fold cross-validation for hyper-parameter selection. The computational burden makes K-fold CV challenging in larger problems. The `pense` package combines several heuristics, as outlined below, to make cross-validation a feasible strategy for hyper-parameter selection for adaptive PENSE.

Throughout this section it is assumed that the penalty loadings $\boldsymbol{\omega} \in \mathbb{R}_+^p$ are fixed. For adaptive PENSE, this means that both the initial estimate $\tilde{\boldsymbol{\beta}}$ and the exponent ζ are fixed. If ζ is to be chosen based on the available data as well, the steps detailed below can be repeated for different penalty loadings.

Hyper-parameter selection via K-fold CV relies on suitably standardized data to ensure comparability of penalization levels across CV folds. To simplify standardization within each individual CV fold, the entire data set $(\mathbf{y}, \mathbf{X})$ is standardized as well. The goal of standardization is to make penalization levels more comparable between individual CV folds and the full data set, requiring the S-loss function, $\mathscr{L}_{\text{S}}$, to be on a standardized scale. Every predictor is centered and scaled by its univariate location and scale estimated as

$$\hat{\mu}_j = \arg\min_{\mu} \hat{\sigma}_{\text{M}}(\mathbf{x}_{\cdot j} - \mu) \quad \text{and} \quad \hat{\sigma}_j = \hat{\sigma}_{\text{M}}(\mathbf{x}_{\cdot j} - \hat{\mu}_j) \quad \text{for } j = 1, \dots, p.$$

Similarly, the S-estimate of location of the observed responses $\hat{\mu}_y = \arg\min_{\mu} \hat{\sigma}_{\text{M}}(\mathbf{y} - \mu)$

is used to center the response. With these estimates of location and scale the data is standardized by

$$\tilde{\mathbf{y}} = \mathbf{y} - \hat{\mu}_y \quad \text{and} \quad \tilde{\mathbf{X}} = \left(\frac{\mathbf{x}_{\cdot 1} - \hat{\mu}_1}{\hat{\sigma}_1}, \ldots, \frac{\mathbf{x}_{\cdot p} - \hat{\mu}_p}{\hat{\sigma}_p} \right). \tag{6.19}$$

An estimate $\tilde{\boldsymbol{\theta}}$ computed on the standardized data can be un-standardized according to

$$\hat{\boldsymbol{\beta}} = \operatorname{diag}\left(1/\hat{\sigma}_1, \ldots, 1/\hat{\sigma}_p\right) \tilde{\boldsymbol{\beta}} \quad \text{and} \quad \hat{\mu} = \tilde{\mu} - \hat{\mu}_y + \left(\hat{\mu}_1, \ldots, \hat{\mu}_p\right) \hat{\boldsymbol{\beta}}. \tag{6.20}$$

To avoid introducing distracting notation the subsequent steps assume that the data set $(\mathbf{y}, \mathbf{X})$ is standardized.

For given penalty loadings $\boldsymbol{\omega}$, the goal is to select a tuple (α^*, λ^*) of hyper-parameters leading to good prediction performance of the estimate. As long as $0 < \alpha < 1$, the effect of the α parameter on the estimate and hence the prediction performance is small compared to the effect of the penalization level λ. Furthermore, α, the balance between the L_1 and L_2 penalties, can be more intuitively interpreted. Therefore, it is usually sufficient to consider only a small number of different values for α. In the following, the set of values considered for the parameter α is denoted by $\mathscr{A}$, which typically consists of only a few values, e.g., $\mathscr{A} = \{1/3, 2/3, 1\}$. Since variable selection is of primary concern, $\mathscr{A}$ usually does not contain 0. While the adaptive PENSE objective function is smooth in α, the coarse grid $\mathscr{A}$ does not emit any gains in computational performance when sharing information across values in $\mathscr{A}$. Therefore, prediction performance of adaptive PENSE at different hyper-parameter settings is estimated independently for each value of α in $\mathscr{A}$ according to the following steps.

Step 1 (defining a grid of penalization levels): The penalization level λ has a much more pronounced yet subtle effect on the adaptive PENSE estimates than the hyper-parameter α. It is therefore important to cover a wide range of penalization levels over a fine-grained grid. Going beyond a penalization level where all coefficient estimates are necessarily 0 is pointless but determining this penalization level is difficult due to the non-convex objective function as discussed in Section 3.5.1. The results in Section 3.5.1 can be extended to show that for given α and penalty loadings $\boldsymbol{\omega}$, the smallest penalization level

for which $\mathbf{0}_p$ is a stationary point of the adaptive PENSE objective function is given by

$$\widetilde{\lambda}_{\text{AS}} = \frac{1}{n\omega_j\alpha} \max_{j=1,\ldots,p} \left| \sum_{i=1}^{n} w_i^2(\mathbf{y} - \hat{\mu}_y)(y_i - \hat{\mu})x_{ij} \right|. \tag{6.21}$$

with $\hat{\mu}_y = \arg\min_\mu \hat{\sigma}_{\text{M}}(\mathbf{y} - \mu)$ and weights $w_i(\mathbf{y} - \hat{\mu}_y)$ as defined in (3.3). For standardized data, $\hat{\mu}_y = 0$.

It is typically not necessary to consider penalization levels greater than $\widetilde{\lambda}_{\text{AS}}$. The `pense` package spans a logarithmically-spaced grid of Q penalization levels from $\widetilde{\lambda}_{\text{AS}}$ to $10^{-3}\alpha\widetilde{\lambda}_{\text{AS}}$, denoted by $\mathscr{Q} = \{\lambda_1, \ldots, \lambda_Q\}$. It is important to note that the penalization levels are in decreasing order, i.e., $\lambda_q > \lambda_{q+1}$, for all $q = 1, \ldots, Q-1$.

Step 2 (defining CV folds): With α and $\mathscr{Q}$ fixed, the n observations are randomly split into K cross-validation folds. The K CV folds are defined through randomly generated "folds", i.e., disjoint index sets $\mathscr{S}^{(k)} \subset \{1, \ldots, n\}$, $k = 1, \ldots, K$ of roughly equal size which include all observations, i.e. $\bigcup_{k=1}^{K} \mathscr{S}^{(k)} = \{1, \ldots, n\}$.

Step 3 (cross-validation): For every single fold $\mathscr{S}^{(k)}$, the training data is defined by

$$\mathbf{y}^{(k)} = \left(y_i \colon i \notin \mathscr{S}^{(k)}\right) \qquad\qquad \mathbf{X}^{(k)} = \left(\mathbf{x}_i \colon i \notin \mathscr{S}^{(k)}\right)^\intercal$$

and contains $n - |\mathscr{S}^{(k)}|$ observations.

With the reduced number of observations in the training data, the robustness parameter δ needs to be adjusted. Given δ fixed beforehand, at most $\lfloor n\delta \rfloor$ observations may be contaminated. Since the training data is a random subset of the entire data set, all contaminated observations may be contained in this particular subset. To guard against this potentially increased proportion of contamination, the parameter needs to be adjusted to $\delta^{(k)} = \lfloor n\delta \rfloor / (n - |\mathscr{S}^{(k)}|)$. In other words, cross-validation effectively decreases the maximum breakdown point attainable by robust estimators to $\delta \leq 0.5(n - \max_{k=1,\ldots,K} |\mathscr{S}^{(k)}|)$.

Step 3.1 (standardizing training data): The training data is standardized according to (6.19), with the location and scale estimates $\hat{\sigma}_j$, $\hat{\mu}_j$, and $\hat{\mu}_y$ estimated on the training data. The fixed penalization levels $\mathscr{Q}$ have approximately the same effect on the adaptive PENSE estimate computed on the standardized training data as if computed on the entire standardized data set.

Step 3.2 (computing the regularization path): The grid of penalization levels typi-

cally contains many different values and computing adaptive PENSE solutions for each of these levels is computationally the most demanding step. To ensure K-fold CV is feasible even for larger data sets, the `pense` package optimizes computing all estimates along this "regularization path", i.e., for all $\lambda \in \mathscr{Q}$ where α and $\boldsymbol{\omega}$ are fixed, as detailed in Algorithm 4.

Before the regularization path can be computed, initial estimates $\mathscr{T}$ are obtained according to Section 6.2. It is both unfeasible and unnecessary to compute initial estimates for every penalty level in $\mathscr{Q}$. By default, the `pense` package computes initial estimates for every fifth penalization level, $\mathscr{Q}_I = \{\lambda_1, \lambda_6, \lambda_{11}, \dots\}$. Many initial estimates do not lead to a good local optimum or lead to the same optimum found with a different starting point. To avoid squandering computational resources on initial estimates without merit, the `pense` package employs a two-stage strategy for computing the regularization path.

For every penalty level λ_q, the algorithm is separated into two stages: exploration and improvement. In the exploration stage, approximate solutions are computed by the MM algorithm with relaxed numerical tolerance ($\epsilon_{\text{exp}} = 0.1$ by default) and no tightening. To increase chances of finding good local optima, the MM algorithm in the exploration stage is started from every solution found for the previous penalty level λ_{q-1} as well as all initial estimates in $\mathscr{T}$. Using a looser numerical tolerance in the exploration stage, the MM algorithm runs for only a few iterations, reducing the computational burden of exploring all possible starting points.

In the second stage, the MM algorithm is started from each of the M best approximate solutions. In this improvement stage, the MM algorithm runs until convergence to the desired numerical tolerance (by default 10^{-6}) and the best solution is retained for each $\lambda \in \mathscr{Q}$. In both the exploration and improvement stage, solutions are judged by their associated value of the adaptive PENSE objective function. This two-stage approach strikes a balance between vast exploration and feasible computation and is successfully applied for many other robust estimators as well (e.g., Salibián-Barrera and Yohai 2006; Rousseeuw and Van Driessen 2006; Alfons et al. 2013). Empirical results suggest that "good" solutions can be differentiated from "bad" solutions after only a few iterations of the MM algorithm.

The inner loops of Algorithm 4 (on lines 4 and 13) can be efficiently distributed among multiple cores, significantly accelerating computation. The outer loop, however, must be done sequentially as sharing information between subsequent penalization levels improves the likelihood of uncovering good local optima.

Step 3.3 (predicting values): Prediction performance of the coefficient estimates along the regularization path is estimated through the prediction error on the test set in the CV

fold. The coefficient estimates must be un-standardized using (6.20) with location and scale estimates obtained for the training data in step 3.1. The prediction errors from un-standardized estimates $\{\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(Q)}\}$ are then given by

$$e_{i,q} = y_i - \hat{\mu}^{(q)} - \mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}}^{(q)} \quad \text{for all } i \in \mathscr{S}^{(k)},\ q = 1, \ldots, Q.$$

Step 4 (computing estimate of prediction performance): After step 3, each observation $i = 1, \ldots, n$ has Q associated prediction errors; one for every considered penalty level. Prediction performance of adaptive PENSE estimates at each penalization level is estimated by the τ-scale of the prediction errors

$$\hat{\tau}_{\alpha,\lambda_q} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \max\left(c_\tau, \frac{|e_{i,q}|}{\underset{i'=1,\ldots,n}{\text{Median}}\,|e_{i',q}|}\right)^2} \qquad \alpha \in \mathscr{A},\ q = 1, \ldots, Q, \tag{6.22}$$

where efficiency constant $c_\tau = 3$ by default in `pense`.

Step 5 (repeating CV with different splits): The non-convexity of the objective function leads to difficulties for cross-validation, as detailed in Section 3.5. This is underlined by empirical results showing that the CV curve of the prediction performance is typically very rough and unstable; varying whimsically between different cross-validation splits. This is clearly visible in the left panel of Figure 6.8, showing the cross-validated prediction performance of adaptive PENSE using two different CV splits on simulated data alongside prediction performance as estimated on an independent validation set. The individual CV curve roughly match the prediction performance from the validation set, but the curves are capricious. Considering only a single CV curve to determine good hyper-parameters is therefore suboptimal as the location of the minimum is most likely not corresponding to a level of penalization leading to the best prediction performance. When averaging the prediction performance estimated over several replications (i.e., cross-validation splits), the CV curve exhibits a smoother surface as shown in the right panel of Figure 6.8. Therefore, the implementation in the `pense` package repeats steps 2 to 4 R times and averages the prediction performance at every λ_q over these R replications:

$$\bar{\tau}_{\alpha,\lambda_q} = \frac{1}{R}\sum_{r=1}^{R} \hat{\tau}^{(r)}_{\alpha,\lambda_q} \qquad \alpha \in \mathscr{A},\ q = 1, \ldots, Q.$$

Averaging multiple CV replications leads to a smoother CV curve and furthermore allows

for accurate estimation of the variability of the estimated prediction performance at any considered penalty level. This enables a more sensible selection of the hyper-parameters for adaptive PENSE. For a fixed α a commonly employed strategy is to not choose λ_q at which the average prediction performance is minimized, but to rather choose a larger penalization level (i.e., a sparser solution) at which the average prediction performance is statistically "indistinguishable" from the smallest average prediction performance. The `pense` package implements this strategy by allowing the user to specify the multiple of the standard error of the smallest average prediction performance considered "indistinguishable", i.e., a generalization of the "one-standard-error" rule (Hastie et al. 2009). In Figure 6.8(b), for example, the error bars depict one half standard error and the average best prediction performance is achieved with $\lambda \approx 8.8$ (21 non-zero coefficients). Using a sparser coefficient vector estimated at $\lambda \approx 13.2$ (15 non-zero coefficients), leads to very similar prediction performance with fewer selected predictors and lower false-positive rate (the true model in this simulation has 16 non-zero coefficients).

Steps 1 to 5 are performed independently for every $\alpha \in \mathscr{A}$. With multiple replications of CV for each α, selecting good hyper-parameters for PENSE and adaptive PENSE is computationally very taxing. While many steps can be efficiently parallelized onto multiple cores or compute nodes, the two-stage approach for computing the regularization path with Algorithm 4 is important to ensure scalability. Without the optimized algorithms described in this chapter, computation would not be feasible for realistic problem sizes.

6.5 Summary

Computation of adaptive PENSE estimates is challenging yet crucial for successful application. Easing the use of adaptive PENSE and making it available to a large audience, the R package `pense` is published on CRAN, the central system for packages extending R. The design goal of the `pense` package is to make adaptive PENSE a versatile tool and applicable to a wide range of problems.

Non-convexity of the objective function combined with necessary selection of hyper-parameters and possible contamination require several novel or adapted computational optimizations to make adaptive PENSE a method of choice. It turns out that all computations can be decomposed into a series of weighted least-squared adaptive elastic net problems. Each of these subproblems is convex and solvable efficiently. However, because of their sheer number, even these supposedly banausic subproblems require diligent optimizations using

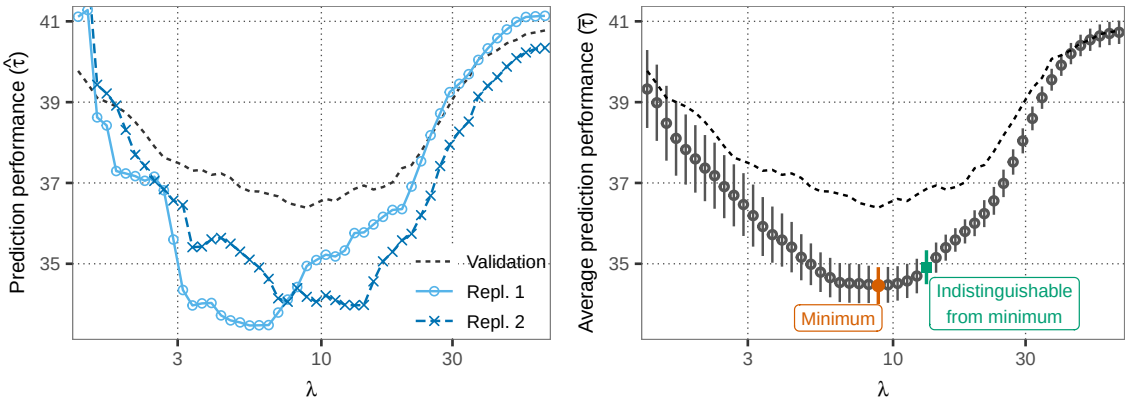
Algorithm 4 Regularization path of adaptive PENSE

Input: Set of penalty levels $\mathscr{Q} = \{\lambda_1, \ldots, \lambda_Q\}$ in decreasing order, set of initial estimates $\mathscr{T}$, maximum number of estimates to improve, $M > 0$, coarse convergence tolerance for exploration $\epsilon_{\text{exp}} > 0$.

Define $\hat{\boldsymbol{\theta}}^{(0)} = \mathbf{0}_{p+1}$.
for $q = 1, \ldots, Q$ **do**
    Initialize an empty set of approximate solutions $\mathscr{B}^{(q)} = \{\}$.
    for $\tilde{\boldsymbol{\theta}} \in \left\{\hat{\boldsymbol{\theta}}^{(q-1)}\right\} \cup \mathscr{T}$ **do**
        Starting the MM algorithm from $\tilde{\boldsymbol{\theta}}$, compute an approximate solution $\hat{\boldsymbol{\theta}}$ using a convergence tolerance of ϵ_{exp}.
        if Set of approximate solutions is not full, i.e., $|\mathscr{B}^{(q)}| < M$ **then**
            Add $\hat{\boldsymbol{\theta}}$ to the set of approximate solutions, $\mathscr{B}^{(q)}$.
        else if $\mathscr{O}(\hat{\boldsymbol{\theta}}; \lambda_q) < \max\{\mathscr{O}(\boldsymbol{\theta}; \lambda_q) \colon \boldsymbol{\theta} \in \mathscr{B}^{(q)}\}$ **then**
            Replace the worst approximate solution in $\mathscr{B}^{(q)}$ by $\hat{\boldsymbol{\theta}}$.
        end if
    end for
    Initialize best optimum as $\hat{\boldsymbol{\theta}}^{(q)} = \mathbf{0}_{p+1}$.
    for $\tilde{\boldsymbol{\theta}} \in B^{(q)}$ **do**
        Starting the MM algorithm from $\tilde{\boldsymbol{\theta}}$, compute a local minimum of the adaptive PENSE objective function, denoted by $\hat{\boldsymbol{\theta}}$.
        if $\mathscr{O}(\hat{\boldsymbol{\theta}}; \lambda_q) < L(\hat{\boldsymbol{\theta}}^{(q)}; \lambda_q)$ **then**
            Update the best optimum to $\hat{\boldsymbol{\theta}}^{(q)} = \hat{\boldsymbol{\theta}}$.
        end if
    end for
end for
Return the set of all solutions, $\{\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(Q)}\}$.

the specific characteristics of the sequence of problems. The `pense` offers three algorithms for weighted LS-adaEN with optimizations to efficiently handle small changes in the data matrix or in the weights. Each of these three algorithms has certain features making them applicable to specific problem sizes and configurations, covering a wide range of problems.

Numerically locating optima of the non-convexity adaptive PENSE objective function necessitates a careful selection of starting points using the EN-PY procedure. Computing EN-PY initial estimates simultaneously for several penalty parameters allows for computational shortcuts. Once these starting points are computed, local optima of the adaptive PENSE objective function can be computed using a minimization-by-majorization (MM) algorithm. I show that the weighted LS-adaEN objective function with properly chosen weights is a useful surrogate function for the adaptive PENSE objective function. Solving a sequence of these weighted LS-adaEN problems leads to a local minimum of the adaptive



(a) CV curve from individual CV splits.

(b) Average prediction performance.

Figure 6.8: Prediction performance of adaptive PENSE ($\alpha = 0.5$) estimated by 100 replications of 7-fold cross-validation on data simulated according to scheme *MS1-MH(-5, 2)* with $n = 100$ and $p = 32$. The black dashed line in both plots shows the prediction error as estimated on an independent validation set. The error bars in the right plot depict half the standard error.

PENSE objective function. Computing a large number of these local minima using different starting points improves the likelihood of finding a global minimum, or at least a local minimum close to the global minimum, unaffected by contamination.

A good choice of the hyper-parameters governing the penalization of the estimates is unknown in practice. Selecting these hyper-parameters therefore usually involves computing adaptive PENSE estimates for many different combinations of the hyper-parameters. As with initial estimates, several computational shortcuts are possible when computing adaptive PENSE for a sequence of hyper-parameters. These optimizations are essential to making computation of adaptive PENSE feasible for realistic problem sizes. Especially because hyper-parameter selection for adaptive PENSE using cross-validation inherently leads to high variance of the estimated prediction performance, requiring several replications of CV, escalating the computational burden. The algorithms and methods implemented in the `pense` package incorporate many optimizations exploiting the characteristics of the adaptive PENSE objective function. These optimizations ensure that adaptive PENSE is computable using reasonable resources for many problems and thus a feasible alternative in most applications.

Chapter 7

Conclusions

This dissertation highlights the inherent challenges arising when considering the possibility of contamination in a sample with many potential predictors but only a limited number of observations. These challenges motivate the development of novel estimators for high dimensional, sparse linear regression models under the presence of contamination with the goal of accurate prediction of the response for a new set of observations and simultaneous identification of a small number of predictors relevant for prediction.

Combining ideas for robust estimation in low-dimensional linear regression models with regularization for variable selection, Chapter 3 proposes the penalized elastic net S-estimator. For robustness of the estimator entails a non-convex objective function, considerable efforts are devoted to guide exploration of the objective function in the quest to locate global minima. The EN-PY procedure is shown to outperform other methods both in terms of quality of the uncovered minima and computational costs. The asymptotic guarantees established for the estimator underline its appropriateness for challenging problems with heavy tailed error distributions and potential contamination in the observed response or predictor values. Data-driven hyper-parameter search is vulnerable to high variance of the performance estimate which is inflated by the presence of contamination and the non-convexity of the objective function. Nevertheless, empirically cross-validation leads to good prediction performance of PENSE, from chimerical scenarios without contamination and well-behaved error terms, to the most challenging situations with heavy-tailed errors and gross contamination.

The PENSE estimator reliably identifies relevant predictors from the large set of available predictors, but theoretical and empirical results expose one shortcoming of the PENSE estimator: insufficient filtering of truly irrelevant predictors. In Chapter 4 I therefore pro-

pose the adaptive PENSE estimator which leverages the PENSE estimator to substantially decrease the number of falsely selected predictors while at the same time retaining the predictive capabilities. Asymptotically, the adaptive PENSE estimator is proven to filter out all irrelevant predictors with high probability, while simultaneously estimating the parameters of the truly relevant predictors with the same efficiency as if the truly relevant predictors were known in advance. This oracle property of the adaptive PENSE estimator, combined with the empirically demonstrated performance even in very challenging scenarios, ascertains reliability and practical advantages of adaptive PENSE.

Analysis of the interplay between sparsity of the true model and contamination of the predictors accentuates the effects of two forms of contamination in the predictors not propagated to the response value: (i) extreme values in predictors with truly non-zero coefficient and (ii) extreme values in truly irrelevant predictors. Prediction performance and variable selection of PENSE is unscathed by contamination (i), while variable selection of non-robust estimators is erratic. Under contamination (ii), on the other hand, it is shown the PENSE estimate is inherently unable to filter out the irrelevant predictors with contaminated values, whereas non-robust methods are more resilient to the effects of these "good" leverage points. Adaptive PENSE combines the best of both worlds, with prediction performance and variable selection unscathed by either form of contamination. Anecdotally, contamination (ii) is very common in practical applications, as the sheer number of irrelevant predictors creates more space for this form of contamination.

Adaptive PENSE's robustness of variable selection and its good prediction performance are germane to meaningful and generalizable scientific results. The utility of adaptive PENSE is demonstrated in a biomarker discovery study with the goal of identifying proteins relevant for predicting cardiac allograft vasculopathy. Adaptive PENSE is estimated to give more accurate predictions using a smaller panel of proteins than other robust or non-robust estimators.

Chapter 5 outlines the problem of residual scale estimation in sparse high-dimensional linear regression models under the presence of contamination. Many proposals for robust regularized regression estimators depend on the availability of an accurate and robust estimate of the residual scale for efficient estimation but also to retain robustness. Theoretical results in low dimensional settings justifying computational shortcuts without sacrificing efficiency are not applicable to regularized M-estimators, entailing a substantial leap of faith when computing M-estimates on possibly contaminated finite samples. I highlight prevalence of severe under- and overestimation of the residual scale in high-dimensional linear

regression, leading to degraded performance of M-estimators. The bias in the scale estimate proves difficult to remove in finite-samples, and strategies for de-biasing proposed for non-robust methods seem unfit for the use with robust estimators. Despite the arguably better performance of regularized M-estimators in less challenging scenarios, the elevated risk of being subjected to the undue influence of contamination, signify more robust alternatives PENSE and adaptive PENSE are to be preferred in practice.

For PENSE and adaptive PENSE to be viable methods for high dimensional data analysis, they need to be readily available in the form of software capable of computing the estimates in a wide range of scenarios. Chapter 6 details adaptations and optimizations of numerical algorithms for use as building blocks in the algorithm devised for computing local minima of the (adaptive) PENSE objective function. Together with an efficient implementation of the EN-PY procedure to guide the search for global minima, (adaptive) PENSE can be efficiently computed for a host of problem sizes. Repeated cross-validation can effectively reduce the high variability of the hyper-parameter search and further improve prediction performance, variable selection, and reliability of the (adaptive) PENSE estimate. With the optimizations developed in Chapter 6, computation of (adaptive) PENSE estimates remains feasible even in high-dimensional settings.

The methods developed in this dissertation gain robustness by down-weighting potentially contaminated observations. An observation is considered contaminated if either the residual or any of it's predictor values is contaminated, following the "casewise" contamination model. With a large number of predictors available in high-dimensional datasets, this approach may lead to problems as even a small number of contaminated values can translate to a large proportion of contaminated observations. Robust methods for the "cellwise" contamination model (Alqallaf et al. 2009), on the other hand, aim at identifying individual values (i.e., cells in the data matrix) with potential contamination and gain robustness by reducing the influence of these cells on the estimation procedure. This strategy is better equipped for high-dimensional datasets, as contamination is not "propagated" from a single value to the entire observation. Methods for the cellwise contamination model, however, are computationally substantially more challenging than PENSE or adaptive PENSE. Importantly, the sparsity assumption imposed in this dissertation alleviates the propagation effect to a certain degree, as aberrant values in the many irrelevant predictors do not pose the same challenges as aberrant values in relevant predictors. In particular adaptive PENSE shows very reliable prediction and variable selection properties in the presence of these forms of contamination, without the need to down-weight affected observations. It would

be nevertheless interesting to investigate a possible combination of techniques used in the cellwise contamination model with adaptive PENSE in future research.

The statistical theory developed for PENSE and adaptive PENSE sheds light on their robustness and asymptotic properties under a general linear regression model. While the considered model covers a wide range of situations, some limitations cannot be ignored. The asymptotic properties of the estimators, for example, are derived under the assumption of i.i.d. errors, which in particular implies that the errors are independent of the predictors and homoscedastic (if F_0 has finite variance). This assumption is sometimes violated in practical applications. Consistency of unregularized S-estimators holds even if these assumptions are violated (Maronna et al. 2019), suggesting that similar extensions may be possible for PENSE and adaptive PENSE. Furthermore, the high breakdown point of the estimators requires a fixed set of hyper-parameters and does not account for any effects of choosing the hyper-parameters based on the potentially contaminated sample. To mitigate the effects of contamination, Chapters 3 and 4 stress the importance of using a robust measure of prediction performance. While empirical results demonstrate the proposed cross-validation scheme selects hyper-parameters which lead to reliable estimates, further analysis of the breakdown point under this scheme would give a more practical assessment of the procedures' robustness towards contamination.

The many facets of contamination in high-dimensional data paired with variable selection and regularized estimation outlined in this dissertation point to several other challenges left for future research. Foremost, low efficiency of the proposed S-estimators in some scenarios suggests room for improvement. Regularized M-estimators are fettered by the high bias in robust estimates of the residual scale as currently available. Building upon the initial study of the problem in this work, grokking the sources of bias in finite samples is crucial to eventual development of appropriate countermeasures and hence more reliable regularized M-estimators. Loh (2018), Fan et al. (2018), and other proposed methods, circumvent the problem of scale estimation altogether by choosing the scaling of the residuals for convex M-estimators from a grid of candidate values, but the theory currently does not adequately support robust estimation under the presence of contamination in the predictors. A potential avenue for future advances is combining the ideas of an adaptive search for appropriate scaling with highly robust regularized estimators. It is of particular interest whether an adaptive search is feasible and reliable under the presence of contaminated predictors. Similarly, other proposals for highly robust estimators for low-dimensional linear regression models can serve as blueprints for robust regularized estimators with higher efficiency than

S-estimators. As the distinct computational advantage of MM-estimators over other highly robust and efficient estimators vanishes in higher dimensions and in presence of a penalty term, alternatives such as the τ estimator (Yohai and Zamar 1988), may be more practicable. It remains for future research to see whether these approaches can be adapted to the sparse linear regression model while retaining efficiency and robustness.

With the proliferation of data seen in recent history, sparse linear regression models are ubiquitous in many areas. The demonstrated reliability of the proposed estimators combined with an efficient implementation for the software environment R, available from `https://cran.r-project.org/package=pense`, will improve generalizability of predictive models and aid future scientific discoveries.

Bibliography

Akaike, H. (1974). "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6, pp. 716–723.

Alfons, A., C. Croux, and S. Gelper (2013). "Sparse least trimmed squares regression for analyzing high-dimensional large data sets". In: *The Annals of Applied Statistics* 7.1, pp. 226–248.

Alqallaf, F., S. Van Aelst, V. J. Yohai, and R. H. Zamar (2009). "Propagation of outliers in multivariate data". In: *The Annals of Statistics* 37.1, pp. 311–331.

Anderson, E. et al. (1999). *LAPACK Users' Guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics. ISBN: 9780898719604.

Arslan, O. (2016). "Penalized MM regression estimation with L_γ penalty: a robust version of bridge regression". In: *Statistics* 50.6, pp. 1236–1260.

Bagirov, A. M., L. Jin, N. Karmitsa, A. Al Nuaimat, and N. Sultanova (2013). "Subgradient method for nonconvex nonsmooth optimization". In: *Journal of Optimization Theory and Applications* 157.2, pp. 416–435.

Belloni, A., V. Chernozhukov, and L. Wang (2011). "Square-root lasso: pivotal recovery of sparse signals via conic programming". In: *Biometrika* 98.4, pp. 791–806.

Bertsekas, D. P. (1982). *Constrained optimization and Lagrange multiplier methods*. New York, NY: Academic Press. ISBN: 9780120934805.

Bertsimas, D., A. King, and R. Mazumder (2016). "Best subset selection via a modern optimization lens". In: *The Annals of Statistics* 44.2, pp. 813–852.

Boyd, S., S. Boyd, and L. Vandenberghe (2004). *Convex Optimization*. Cambridge, MA: Cambridge University Press. ISBN: 9780521833783.

Bühlmann, P. and S. van de Geer (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Berlin Heidelberg: Springer.

Chang, L., S. Roberts, and A. Welsh (2018). "Robust lasso regression using Tukey's biweight criterion". In: *Technometrics* 60.1, pp. 36–47.

Chatterjee, S. and J. Jafarov (2015). "Prediction error of cross-validated lasso". In: *ArXiv e-prints*, arXiv:1502.06291.

Chen, Z., J. Fan, and R. Li (2018). "Error variance estimation in ultrahigh dimensional additive models". In: *Journal of the American Statistical Association* 113.512, pp. 315–327.

Clarke, F. (1990). *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics. ISBN: 9781611971309.

Cohen Freue, G. V., D. Kepplinger, M. Salibián-Barrera, and E. Smucler (2019). "Robust elastic net estimators for variable selection and identification of proteomic biomarkers". In: *Annals of Applied Statistics* 13.4, pp. 2065–2090.

Davies, L. (1990). "The asymptotics of S-estimators in the linear regression model". In: *The Annals of Statistics* 18.4, pp. 1651–1675.

Davies, P. L. and U. Gather (2005). "Breakdown and groups". In: *The Annals of Statistics* 33.3, pp. 977–1035.

Davis, D. and W. Yin (2017). "Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions". In: *Mathematics of Operations Research* 42.3, pp. 783–805.

Deng, W. and W. Yin (2016). "On the global and linear convergence of the generalized alternating direction method of multipliers". In: *Journal of Scientific Computing* 66.3, pp. 889–916.

Dicker, L. H. (2014). "Variance estimation in high-dimensional linear models". In: *Biometrika* 101.2, pp. 269–284.

Donoho, D. L. and P. J. Huber (1982). "The notion of breakdown point". In: *A Festschrift For Erich L. Lehmann*. Ed. by P. J. Bickel, D. K., and J. Hodges. CRC Press, pp. 157–184.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). "Least angle regression". In: *The Annals of Statistics* 32.2, pp. 407–499.

Fan, J., S. Guo, and N. Hao (2012). "Variance estimation using refitted cross-validation in ultrahigh dimensional regression". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.1, pp. 37–65.

Fan, J., Q. Li, and Y. Wang (2017). "Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.1, pp. 247–265.

Fan, J. and R. Li (2001). "Variable selection via nonconcave penalized likelihood and its oracle properties". In: *Journal of the American Statistical Association* 96.456, pp. 1348–1360.

Fan, J., H. Liu, Q. Sun, and T. Zhang (2018). "I-LAMM for sparse learning: simultaneous control of algorithmic complexity and statistical error". In: *The Annals of Statistics* 46.2, pp. 814–841.

Fan, J. and H. Peng (2004). "Nonconcave penalized likelihood with a diverging number of parameters". In: *The Annals of Statistics* 32.3, pp. 928–961.

Fan, J., W. Wang, and Z. Zhu (2016). "A shrinkage principle for heavy-tailed data: high-dimensional robust low-rank matrix recovery". In: *arXiv e-prints*, arXiv:1603.08315.

Fan, J., L. Xue, and H. Zou (2014). "Strong oracle optimality of folded concave penalized estimation". In: *The Annals of Statistics* 42.3, pp. 819–849.

Friedman, J., T. Hastie, and R. Tibshirani (2010). "Regularization paths for generalized linear models via coordinate descent". In: *Journal of Statistical Software, Articles* 33.1, pp. 1–22.

Gentle, J. E. (2007). *Matrix Algebra: Theory, Computations, and Applications in Statistics*. 2nd edition. Springer Texts in Statistics. New York, NY: Springer. ISBN: 9780387708737.

Gill, P. E., G. H. Golub, W. Murray, and M. A. Saunders (1974). "Methods for modifying matrix factorizations". In: *Mathematics of Computation* 28.126, pp. 505–535.

Hampel, F. R. (1975). "Beyond location parameters: robust concepts and methods". In: *Bulletin of the International Statistical Institute* 46.1, pp. 375–382.

— (1974). "The influence curve and its role in robust estimation". In: *Journal of the American Statistical Association* 69.346, pp. 383–393.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*. 2nd edition. New York, NY: Springer.

Hastie, T., R. Tibshirani, and R. J. Tibshirani (2017). "Extended comparisons of best subset selection, forward stepwise selection, and the lasso". In: *arXiv e-prints*, arXiv:1707.08692.

He, B. and X. Yuan (2015). "On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers". In: *Numerische Mathematik* 130.3, pp. 567–577.

Hirose, K., S. Tateishi, and S. Konishi (2013). "Tuning parameter selection in sparse regression modeling". In: *Computational Statistics & Data Analysis* 59, pp. 28–40.

Homrighausen, D. and D. J. McDonald (2016). "Risk-consistency of cross-validation with lasso-type procedures". In: *ArXiv e-prints*.

Homrighausen, D. and D. J. McDonald (2018). "A study on tuning parameter selection for the high-dimensional lasso". In: *Journal of Statistical Computation and Simulation* 88.15, pp. 2865–2892.

Hössjer, O. (1992). "On the optimality of S-estimators". In: *Statistics & Probability Letters* 14.5, pp. 413–419.

Huber, P. J. and E. M. Ronchetti (2009). *Robust Statistics*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, Inc.

Jojic, V., S. Saria, and D. Koller (2011). "Convex envelopes of complexity controlling penalties: the case against premature envelopment". In: *Proceedings of the Conference on Artificial Intelligence and Statistics* 15, pp. 399–406.

Khan, J. A., S. V. Aelst, and R. H. Zamar (2007). "Robust linear model selection based on least angle regression". In: *Journal of the American Statistical Association* 102.480, pp. 1289–1299.

Kim, J. and D. Pollard (1990). "Cube root asymptotics". In: *Annals of Statistics* 18.1, pp. 191–219.

Lange, K. (2016). *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics. ISBN: 9781611974409.

Lehmann, E. and G. Casella (2003). *Theory of Point Estimation*. Springer Texts in Statistics. New York, NY: Springer. ISBN: 9780387985022.

Lin, D. et al. (2013). "Plasma protein biosignatures for detection of cardiac allograft vasculopathy". In: *The Journal of Heart and Lung Transplantation* 32.7, pp. 723–733.

Loh, P.-L. (2017). "Statistical consistency and asymptotic normality for high-dimensional robust M-estimators". In: *The Annals of Statistics* 45.2, pp. 866–896.

— (2018). "Scale calibration for high-dimensional robust regression". In: *arXiv e-prints*.

Mammen, E. (1996). "Empirical process of residuals for high-dimensional linear models". In: *The Annals of Statistics* 24.1, pp. 307–335.

Mandelbrot, B. (1960). "The Pareto-Lévy law and the distribution of income". In: *International Economic Review* 1.2, pp. 79–106.

Maronna, R., D. Martin, V. Yohai, and M. Salibián-Barrera (2019). *Robust Statistics: Theory and Methods (with R)*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, Inc. ISBN: 9781119214670.

Maronna, R. and V. J. Yohai (2010). "Correcting MM estimates for "fat" data sets". In: *Computational Statistics & Data Analysis* 54, pp. 3168–3173.

Maronna, R. A. and R. H. Zamar (2002). "Robust estimates of location and dispersion for high-dimensional datasets". In: *Technometrics* 44.4, pp. 307–317.

Maronna, R. A. (2011). "Robust ridge regression for high-dimensional data". In: *Technometrics* 53.1, pp. 44–53.

Mehta, N. U. and S. T. Reddy (2015). "Role of hemoglobin/heme scavenger protein hemopexin in atherosclerosis and inflammatory diseases." In: *Current Opinion in Lipidology* 26.5, pp. 384–387.

Mei, S., Y. Bai, and A. Montanari (2018). "The landscape of empirical risk for nonconvex losses". In: *The Annals of Statistics* 46.6A, pp. 2747–2774.

Mendes, B. and D. E. Tyler (1996). "Constrained M-estimation for regression". In: *Robust Statistics, Data Analysis, and Computer Intensive Methods: In Honor of Peter Huber's 60th Birthday*. Ed. by H. Rieder. New York, NY: Springer, pp. 299–320. ISBN: 9781461223801.

Neve, A., F. P. Cantatore, N. Maruotti, A. Corrado, and D. Ribatti (2014). "Extracellular matrix modulates angiogenesis in physiological and pathological conditions." In: *Biomed Research International* 2014, p. 756078.

Parikh, N. and S. Boyd (2014). "Proximal algorithms". In: *Foundations and Trends® in Optimization* 1.3, pp. 127–239.

Peña, D. and V. J. Yohai (1999). "A fast procedure for outlier diagnostics in large regression problems". In: *Journal of the American Statistical Association* 94.446, pp. 434–445.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.

Reid, S., R. Tibshirani, and J. Friedman (2016). "A study of error variance estimation in lasso regression". In: *Statistica Sinica* 26, pp. 35–67.

Rockafellar, R. (1970). *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Ewing, NJ: Princeton University Press. ISBN: 9780691015866.

Rousseeuw, P. J. (1984). "Least median of squares regression". In: *Journal of the American Statistical Association* 79.388, pp. 871–880.

Rousseeuw, P. J. and A. M. Leroy (1987). *Robust Regression and Outlier Detection*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, Inc. ISBN: 0471852333.

Rousseeuw, P. J. and K. Van Driessen (2006). "Computing LTS regression for large data sets". In: *Data Mining and Knowledge Discovery* 12.1, pp. 29–45.

Rousseeuw, P. J. and V. J. Yohai (1984). "Robust regression by means of S-estimators". In: *Robust and Nonlinear Time Series Analysis*. New York, NY: Springer, pp. 256–272. ISBN: 9781461578215.

Salibián-Barrera, M. and V. J. Yohai (2006). "A fast algorithm for S-regression estimates". In: *Journal of Computational and Graphical Statistics* 15.2, pp. 414–427.

Schmauss, D. and M. Weis (2008). "Cardiac allograft vasculopathy". In: *Circulation* 117.16, pp. 2131–2141.

Schwarz, G. (1978). "Estimating the dimension of a model". In: *The Annals of Statistics* 6.2, pp. 461–464.

Shor, N. (1985). *Minimization Methods for Non-Differentiable Functions*. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer. ISBN: 9783540127635.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2011). "Regularization paths for Cox's proportional hazards model via coordinate descent". In: *Journal of Statistical Software* 39.5, pp. 1–13.

Smucler, E. (2019). "Asymptotics for redescending M-estimators in linear models with increasing dimension". In: *Statistica Sinica* 29, pp. 1065–1081.

Smucler, E. and V. J. Yohai (2017). "Robust and sparse estimators for linear regression models". In: *Computational Statistics & Data Analysis* 111.C, pp. 116–130.

Sun, Q., W.-X. Zhou, and J. Fan (2019). "Adaptive Huber regression". In: *Journal of the American Statistical Association* 115.529, pp. 254–265.

Sun, T. and C.-H. Zhang (2012). "Scaled sparse linear regression". In: *Biometrika* 99.4, pp. 879–898.

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 58.1, pp. 267–288.

Tibshirani, R., J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani (2012). "Strong rules for discarding predictors in lasso-type problems". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 74.2, pp. 245–266.

Tibshirani, R. J. and S. Rosset (2019). "Excess optimism: how biased is the apparent error of an estimator tuned by SURE?" In: *Journal of the American Statistical Association* 114.526, pp. 697–712.

Tomioka, R., T. Suzuki, and M. Sugiyama (2011). "Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation". In: *Journal of Machine Learning Research* 12, pp. 1537–1586.

Van de Geer, S. and P. Müller (2012). "Quasi-likelihood and/or robust estimation in high dimensions". In: *Statistical Science* 27.4, pp. 469–480.

Van der Vaart, A. and J. Wellner (1996). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. New York, NY: Springer. ISBN: 9780387946405.

Vinchi, F., L. De Franceschi, A. Ghigo, T. Townes, J. Cimino, L. Silengo, E. Hirsch, F. Altruda, and E. Tolosano (2013). "Hemopexin therapy improves cardiovascular function by preventing heme-induced endothelial toxicity in mouse models of hemolytic diseases." In: *Circulation* 127.12, pp. 1317–1329.

Wang, H. and R. Li (2007). "Tuning parameter selectors for the smoothly clipped absolute deviation method". In: *Biometrika* 94.3, pp. 553–568.

Watkins, D. S. (2002). *Fundamentals of Matrix Computations*. 2nd edition. New York, NY: John Wiley & Sons, Inc.

Yang, T. (2017). "Adaptive robust methodology for parameter estimation and variable selection". PhD thesis. Clemson, SC: Clemson University. ISBN: 9780355344769.

Yohai, V., R. J. Maronna, D. Martin, G. Brownson, K. Konis, and M. Salibián-Barrera (2019). *RobStatTM: Robust Statistics: Theory and Methods*. R package version 1.0.0.

Yohai, V. J. (1985). *High breakdown point and high efficiency robust estimates for regression*. Tech. rep. 66. University of Washington.

— (1987). "High breakdown-point and high efficiency robust estimates for regression". In: *The Annals of Statistics* 15.2, pp. 642–656.

Yohai, V. J. and R. H. Zamar (1986). *High breakdown-point estimates of regression by means of the minimization of an efficient scale*. Tech. rep. 84. University of Washington.

— (1988). "High breakdown-point estimates of regression by means of the minimization of an efficient scale". In: *Journal of the American Statistical Association* 83.402, pp. 406–413.

Yohai, V. J. and R. H. Zamar (1997). "Optimal locally robust M-estimates of regression". In: *Journal of Statistical Planning and Inference* 64.2, pp. 309–323.

Yu, G. and J. Bien (2019). "Estimating the error variance in a high-dimensional linear model". In: *Biometrika* 106.3, pp. 533–546.

Zhang, C.-H. and T. Zhang (2012). "A general theory of concave regularization for high-dimensional sparse estimation problems". In: *Statistical Science* 27.4, pp. 576–593.

Zhao, Y., J. Chen, J. M. Freudenberg, Q. Meng, null null, D. K. Rajpal, and X. Yang (2016). "Network-based identification and prioritization of key regulators of coronary artery disease loci". In: *Arteriosclerosis, Thrombosis, and Vascular Biology* 36.5, pp. 928–941.

Zou, H. (2006). "The adaptive lasso and its oracle properties". In: *Journal of the American Statistical Association* 101.476, pp. 1418–1429.

Zou, H. and T. Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2, pp. 301–320.

Zou, H. and R. Li (2008). "One-step sparse estimates in nonconcave penalized likelihood models". In: *The Annals of Statistics* 36.4, pp. 1509–1533.

Zou, H. and H. H. Zhang (2009). "On the adaptive elastic-net with a diverging number of parameters". In: *The Annals of Statistics* 37.4, pp. 1733–1751.

Appendix A

Simulation Settings

A.1 Data-Generation Schemes

The p-dimensional predictors, $\mathbf{x}_i$, $i = 1, \ldots, n$ are independent realizations of a p-dimensional random variable X from a multivariate t distribution with 4 degrees of freedom. The correlation structure among the predictors can be one of the following.

Correlation structure 1 [AR(1)]: Exponential decay of the correlation between predictors according to their "distance", $\text{Cor}(\mathcal{X}_j, \mathcal{X}_{j'}) = \rho^{|j-j'|}$, for $j, j' = 1, \ldots, p$. The parameter $0 \leq \rho \leq 1$ determines the general strength of the correlation.

Correlation structure 2 [equal correlation]: All predictors are equally correlated, $\text{Cor}(\mathcal{X}_j, \mathcal{X}_{j'}) = \rho$ for all $j, j' = 1, \ldots, p$, $j \neq j'$.

The response values y_i, $i = 1, \ldots, n$ are generated by a linear combination of the first s predictors:

$$y_i = u_i + \sum_{j=1}^{s} x_{ij}, \quad i = 1, \ldots, n. \tag{A.1}$$

The residuals u_i are scaled versions of raw residuals $\tilde{u}_i$. These unscaled $\tilde{u}_i$ are independent realizations of a random variable $\mathcal{U}$ following a central stable distribution (Mandelbrot 1960) with varying stability parameter α:

LT light-tailed stable distribution with tail parameter $\alpha = 2$, i.e., a Standard Normal distribution,

ML moderate- to light-tailed table distribution with stability parameter $\alpha = 1.66$,

MH moderate to heavy-tailed stable distribution with stability parameter $\alpha = 1.33$,

HT heavy-tailed stable distribution with stability parameter $\alpha = 1$, i.e., a Cauchy distribution.

The raw residuals $\tilde{u}_i$ are scaled to attain a certain proportion of variance explained (PVE) by the true linear regression model (A.1):

$$u_i = \sqrt{\frac{1-\nu}{\nu}} \frac{\tilde{u}_i \hat{\sigma}_0}{\hat{\tau}_{\tilde{u}}}$$

where

$$\begin{aligned} \hat{\tau}_{\tilde{u}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n} \max\left(3, \frac{|\tilde{u}_i|}{\underset{i'=1,\ldots,n}{\text{Median}}\,|\tilde{u}_i|}\right)^2} \\ \hat{\sigma}_0 &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\sum_{j=1}^{s} x_{ij} - \frac{1}{n}\sum_{i'=1}^{n}\sum_{j'=1}^{s} x_{i'j'}\right)^2}. \end{aligned} \tag{A.2}$$

This definition of PVE uses a robust measure of spread of the error terms because of the considered error distributions, only the light-tailed Normal distribution has finite variance. Unless otherwise specified, data is generated with $\nu = 0.25$, i.e., the true model explains about 25% of the observed variance in y_i.

Contamination is artificially introduced in $0 \leq n_c < n$ observations. Contaminated observations are generated by a different linear model with strong signal and have high leverage by replacing some predictor values with more extreme values. Usually $n_c = \lfloor n/4 \rfloor$, i.e., 25% contamination, unless otherwise specified.

Leverage points are introduced by contaminating $q = \log_2(p)$ predictors. The indices of contaminated predictors are sampled non-uniformly without replacement from $\{1, \ldots, p\}$ to increase the chances of active predictors being contaminated. This is done by first sampling $q_{\mathcal{A}}$ from a discrete uniform distribution over $\{\max(0, q+s-p), \ldots, \min(q, s)\}$. Then, $q_{\mathcal{A}}$ indices are sampled uniformly without replacement from $\{1, \ldots, s\}$ and $q - q_{\mathcal{A}}$ are sampled uniformly without replacement from $\{s+1, \ldots, p\}$, denoting the sampled indices by $\mathcal{J}_{\mathcal{A}}$ and $\mathcal{J}_{\mathcal{A}^C}$, respectively. The values of these contaminated predictors are replaced by

$$x_{ij} = x_{ij}\sqrt{k_1 \frac{\max\limits_{i'=1,\ldots,p} d_{i'}^2}{d_i^2}} \qquad i = 1, \ldots, n_c,\ j \in \mathcal{J}_{\mathcal{A}} \cup \mathcal{J}_{\mathcal{A}^C} \tag{A.3}$$

where d_i^2 is the squared Mahalanobis distance of the i-th observation, relative to the empirical covariance matrix of the predictors $\mathcal{J}_\mathcal{A} \cup \mathcal{J}_{\mathcal{A}^C}$, estimated over the uncontaminated observations. The placement of the leverage points and thus the severity of leverage is controlled by the parameter k_l which can take values $k_\text{l} \in \{2, 4, 8, 16\}$, corresponding to low, moderate, high, and extreme leverage, respectively.

The response values of the n_c contaminated observations are determined by the q contaminated predictors

$$y_i = u_i + \sum_{j \in \mathcal{J}_\mathcal{A} \cup \mathcal{J}_{\mathcal{A}^C}} k_\text{v} x_{ij} \qquad i = 1, \ldots, n_c, \tag{A.4}$$

where k_v determines the magnitude of the residuals, relative to the true model, and takes values in $\{-2, -1, 0, 3, 7\}$. The larger the difference $|k_\text{v} - 1|$, the more extreme the contamination. In case of contamination, the scale estimates in (A.2) are computed only from the $n - n_c$ uncontaminated observations.

A.1.1 Short-Hand Notation

Data generation schemes are referenced throughout the text according to a short-hand notation as explained in Figure A.1. The short-hand notation consists of four parts. The first two letters specify the sparsity of the true model, i.e., the number of truly active predictors as a function of p, followed by a number identifying the correlation structure among the p predictors. The third part consists of one to two characters denoting the error distribution in terms of the weight of tails. The last part specifies the parameters for contamination. If "(—)", the generated data does not contain contaminated observations, while two numbers in parentheses specify k_v, the parameter for contaminating the model according to (A.4), and k_l, the parameter for contaminating the predictors according to (A.3), in that order. The last part can also be "*", meaning that several combinations of contamination parameters are considered.

The short-hand notation does not specify the dimensions of the generated data, n and p. If not specified otherwise, the data is generated such that the true model explains 25% of the observed in y_i, i.e., $\nu = 0.25$. If the last part of the notation is given, 25% of the observations are contaminated, unless otherwise given in the text.

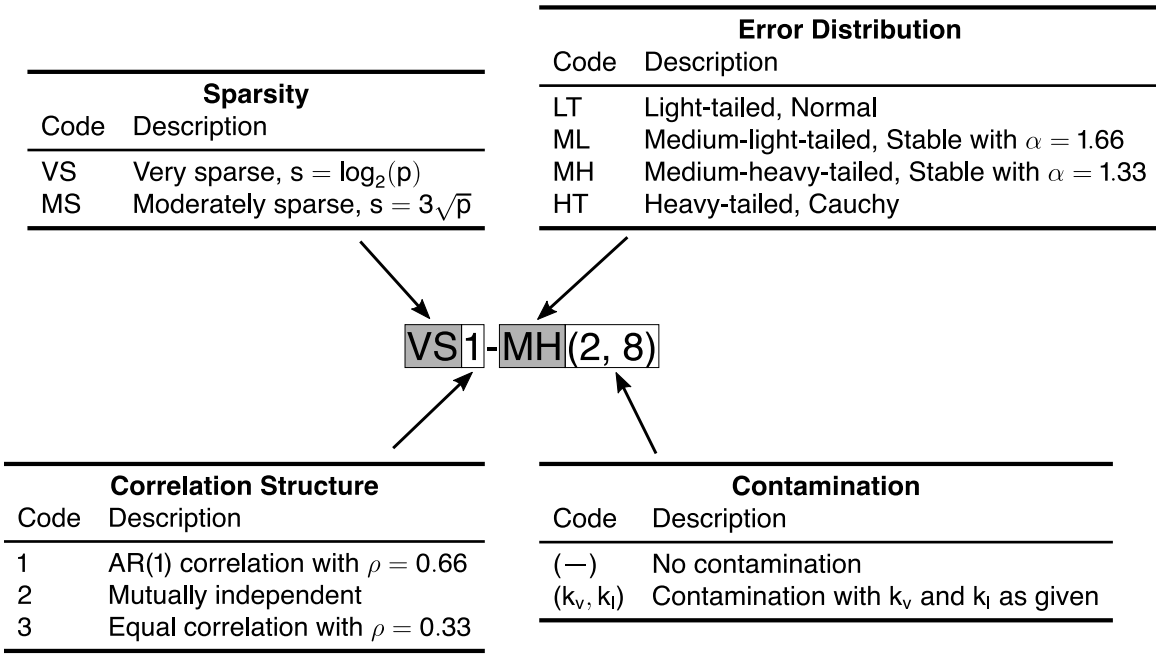


Figure A.1: Short-hand notation for data generation schemes.

A.2 Comparison of Initial Estimates

To compare the performance of initial estimates in Section 3.2.3, data sets of size $n = 100$ and $p = 16$ are generated according scheme *VS1-LT**. The proportion of variance explained is 25% or 50%. For contamination, all combinations of $k_l \in \{1, 2, 4, 8\}$ and $k_v \in \{-2, -1, 0, 3, 7\}$ are considered. Combined with the scenarios of no contamination, this leads to a total of 42 scenarios.

For each of the two scenarios without contamination, 250 data sets are generated, while for scenarios with contamination 50 data sets each are generated. On each of the 2500 data sets, the PENSE estimate is computed over a grid $\mathscr{Q}$ comprising 50 log-spaced penalization levels. At 10 log-spaced penalization levels, $\mathscr{Q}_I$ (spanning the same range as $\mathscr{Q}$), the EN-PY estimates are computed. All of these estimates are used to initialize the PENSE algorithm for each of the 50 values in $\mathscr{Q}$ to find the best local minimum.

In the process of computing the EN-PY estimates, a total of K LS-EN estimates are computed. To make the computational demand for the EN-PY estimator and the random subsampling strategy comparable, a total of $\lceil K/10 \rceil$ random subsamples are taken for the random subsampling strategy. For each of these random subsamples, the LS-EN estimates are computed over the same grid $\mathscr{Q}_I$ as used for EN-PY. All of the K initial estimates are then used to initialize the PENSE algorithm, similar to the EN-PY initial estimates.

A.3 Numerical Experiments for PENSE and Adaptive PENSE

Numerical experiments comparing PENSE and adaptive PENSE to competing methods in Sections 3.6 and 4.4 a consider a large number of scenarios following the data generation detailed in Section A.1. Specifically, data is generated according to data generation schemes *VS1-** and *MS1-**, with $\nu = 0.25$ and varying number of observations and predictors. For $n = 100$ observations $p \in \{16, 32, 64, 128\}$, while for $n = 400$, the number of predictors is either $p = 32$ or $p = 64$. In scenarios with contamination, 25% of observations are affected with leverage parameter k_{l} fixed at 8 and vertical outlier positions $k_{\text{v}} \in \{-2, -1, 0, 3, 7\}$.

PENSE and adaptive PENSE estimates are computed using the `pense` R package available from CRAN and detailed in Chapter 6. MM-LASSO is computed using the code from `https://github.com/esmucler/mmlasso`, implementing the originally algorithm proposed in Smucler and Yohai (2017). Cross-validation for (adaptive) PENSE, in particular standardizing the data and adjusting the robustness parameter δ in the CV folds, is done according to Section 6.4. To ensure computational feasibility in this large-scale simulation study, cross-validation for hyper-parameter selection is performed only a single time for all considered estimates. The reported performance metrics are therefore likely underestimating the true performance of the estimators, albeit all methods should be equally affected.

Appendix B

Proofs

B.1 Breakdown Point of PENSE

Recall that the PENSE estimate, $\tilde{\boldsymbol{\theta}}$, computed from the sample $\mathscr{Z} = (\mathbf{y}, \mathbf{X}) = \{(y_i, \mathbf{x}_i) \colon i = 0, \ldots, n\}$ is given by

$$\tilde{\boldsymbol{\theta}} = \underset{\mu,\boldsymbol{\beta}}{\arg\min}\, \mathscr{O}_{\text{S}}(\mu, \boldsymbol{\beta}; \lambda, \alpha, \mathscr{Z}) = \underset{\mu,\boldsymbol{\beta}}{\arg\min}\, \mathscr{L}_{\text{S}}(\mathbf{y}, \mu + \mathbf{X}\boldsymbol{\beta}) + \lambda \Phi_{\text{EN}}(\boldsymbol{\beta}; \alpha).$$

In the following, contaminated samples derived from $\mathscr{Z}$, where $m < n$ out of the n observations are replaced by arbitrary values are denoted by $\widetilde{\mathscr{Z}_m} = (\tilde{\mathbf{y}}_m, \tilde{\mathbf{X}}_m)$.

To prove the finite-sample breakdown point of PENSE, the following lemma from Maronna et al. (2019, p. 184) is essential.

Lemma 1. *Consider any sequence of samples* $\left(\widetilde{\mathscr{Z}_m^{(k)}}\right)_{k\in\mathbb{N}}$ *with individual observation pairs* $(\tilde{y}_i^{(k)}, \tilde{\mathbf{x}}_i^{(k)})$ *and corresponding residuals* $\tilde{r}_i^{(k)} = \tilde{y}_i^{(k)} - \mu^{(k)} - \left(\tilde{\mathbf{x}}_i^{(k)}\right)^\intercal \boldsymbol{\beta}^{(k)}$ *for any sequence of estimates* $(\mu^{(k)}, \boldsymbol{\beta}^{(k)})$.

(i) Let $C = \left\{i : |\tilde{r}_i^{(k)}| \to \infty\right\}$. *If* $\#(C) > n\delta$, *then* $\hat{\sigma}_M(\tilde{\mathbf{r}}^{(k)}) \to \infty$ *for* $k \to \infty$.

(ii) Let $D = \left\{i : |\tilde{r}_i^{(k)}| \text{ is bounded}\right\}$. *If* $\#(D) > n - n\delta$, *then* $\hat{\sigma}_M(\tilde{\mathbf{r}}^{(k)})$ *is bounded.*

With Lemma 1 in place, the proof of the upper and lower bounds in Theorem 2 is done separately. The following proof of the FBP of PENSE first appeared in Cohen Freue et al. (2019) with slightly different notation.

Proof of Theorem 2, bounded from below. Consider an arbitrary sequence of contaminated samples $\left(\widetilde{\mathscr{Z}_m^{(k)}}\right)_{k\in\mathbb{N}}$ with $m \leq m(\delta)$. The goal is to show that the corresponding sequence of

PENSE estimates, $\left(\tilde{\boldsymbol{\theta}}^{(k)}\right)_{k\in\mathbb{N}}$, remains bounded. The sequence of residuals of these PENSE estimates is denoted by $\tilde{\mathbf{r}}^{(k)} = \tilde{\mathbf{y}}^{(k)} - \tilde{\mu}^{(k)} - (\tilde{\mathbf{x}}_i^{(k)})^\intercal \tilde{\boldsymbol{\beta}}^{(k)}$.

First, let $\boldsymbol{\theta}^*$ fixed for all k such that $|\mu^*| < \infty$ and $\|\boldsymbol{\beta}^*\|_1 = K_1 < \infty$, which implies also finite L_2 norm of the slope $\|\boldsymbol{\beta}^*\|_2^2 = K_2 < \infty$. For those uncontaminated observations $(y_i, \mathbf{x}_i^\intercal)$ which are also in the contaminated sample $\widetilde{\mathscr{Z}}_m^{(k)}$, the triangle inequality says that the residuals $r_i^{*(k)} = y_i - \mu^* - \mathbf{x}_i^\intercal \boldsymbol{\beta}^*$ are bounded, $|r_i^{*(k)}| < \infty$. Therefore, the number of bounded residuals $\#(D) \geq n - m \geq n - n\delta$ and hence part (ii) of Lemma 1 says that $\hat{\sigma}_{\mathrm{M}}(\mathbf{r}^{*(k)})$ is bounded:

$$\sup_{k\in\mathbb{N}} \hat{\sigma}_{\mathrm{M}}(\mathbf{r}^{*(k)}) < \infty. \tag{B.1}$$

Now suppose that the sequence of slope estimates from PENSE, $\left(\|\tilde{\boldsymbol{\beta}}^{(k)}\|_1\right)_{k\in\mathbb{N}}$ is unbounded. It is important to note that the the sequence estimated intercepts may be bounded or unbounded. The boundedness of the M-scale estimate in B.1 implies there exists a $k_0 \in \mathbb{N}$ such that $\|\tilde{\boldsymbol{\beta}}^{(k_0)}\|_1 > K_1 + \frac{1}{\alpha\lambda} \sup_{k\in\mathbb{N}} \hat{\sigma}_{\mathrm{M}}^2(\mathbf{r}^{*(k)})$ and $\|\tilde{\boldsymbol{\beta}}^{(k_0)}\|_2^2 > K_2$. Thus, for every $k' \geq k_0$,

$$\begin{aligned} \mathscr{O}_{\mathrm{S}}(\tilde{\mu}^{(k')}, \tilde{\boldsymbol{\beta}}^{(k')}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}) &> \hat{\sigma}_{\mathrm{M}}^2(\mathbf{r}^{*(k')}) + \lambda\left(\frac{1-\alpha}{2}K_2 + \alpha K_1\right) + \sup_{k\in\mathbb{N}} \hat{\sigma}_{\mathrm{M}}^2(\mathbf{r}^{*(k)}) \\ &\geq \mathscr{O}_{\mathrm{S}}(\mu^*, \boldsymbol{\beta}^*; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}), \end{aligned} \tag{B.2}$$

contradicting the assumption that $\tilde{\boldsymbol{\theta}}^{(k)}$ minimizes the PENSE objective function. This proves that $\tilde{\boldsymbol{\beta}}^{(k)}$ is bounded for $m \leq m(\delta)$ regardless of $\tilde{\mu}^{(k)}$ being bounded or not. It remains to show that the intercept is bounded as well.

Since $\left(\|\tilde{\boldsymbol{\beta}}^{(k)}\|_1\right)_{k\in\mathbb{N}}$ is bounded, $|y_i - \mathbf{x}_i^\intercal \tilde{\boldsymbol{\beta}}^{(k)}|$ is bounded for the $n-m$ uncontaminated observations $(y_i, \mathbf{x}_i)$ in the contaminated sample $\widetilde{\mathscr{Z}}_m^{(k)}$. Assume now that $|\tilde{\mu}^{(k)}| \to \infty$. Then the residuals of the uncontaminated observations also tend to infinity and hence $\#(C) > n\delta$. According to part (i) of Lemma 1 this implies that $\hat{\sigma}_{\mathrm{M}}(\tilde{\mathbf{r}}^{(k)}) \to \infty$. Therefore, there exists an integer $k_1 \in \mathbb{N}$ such that $\hat{\sigma}_{\mathrm{M}}^2(\tilde{\mathbf{r}}^{(k_1)}) > \sup_{k\in\mathbb{N}} \hat{\sigma}_{\mathrm{M}}^2(\mathbf{r}^{*(k)}) + \lambda\left(\frac{1-\alpha}{2}K_2 + \alpha K_1\right)$. Similar to (B.2), this shows that for all $k' \geq k_1$,

$$\mathscr{O}_{\mathrm{S}}(\tilde{\mu}^{(k')}, \tilde{\boldsymbol{\beta}}^{(k')}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}) \geq \mathscr{O}_{\mathrm{S}}(\mu^*, \boldsymbol{\beta}^*; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}),$$

and hence $\tilde{\boldsymbol{\theta}}^{(k)}$ must be bounded for $m \leq m(\delta)$. □

Proof of Theorem 2, bounded from above. Taking $m > n\delta$ it can be shown that the PENSE

estimate breaks down. Without loss of generality, assume that the first m observations in the contaminated samples $\widetilde{\mathscr{Z}}_m^{(k)}$ are different from the original sample $\mathscr{Z}$. Choosing an arbitrary $\mathbf{x}_0$ with $\|\mathbf{x}_0\|_2 = 1$ and $0 < \nu \leq 1$, it can be shown that for the sequence of contaminated samples $\left(\widetilde{\mathscr{Z}}_m^{(k)}\right)_{k\in\mathbb{N}}$,

$$(\tilde{y}_i^{(k)}, \tilde{\mathbf{x}}_i^{(k)}) = \begin{cases} (k^{\nu+1}, k\mathbf{x}_0) & i \in C \\ (y_i, \mathbf{x}_i) & i \notin C \end{cases},$$

the corresponding sequence of estimates $\left(\tilde{\boldsymbol{\theta}}^{(k)}\right)_{k\in\mathbb{N}}$ can not be bounded.

Assume here that $\tilde{\boldsymbol{\theta}}^{(k)}$ is bounded in norm. As in the proof above the residuals of the uncontaminated observations $|\tilde{r}_i^{(k)}| < \infty$ for $i = m+1, \ldots, n$ and all $k \in \mathbb{N}$. Residuals for contaminated samples, on the other hand, are bounded below by

$$|\tilde{r}_i^{(k)}| \geq k \left| k^{\nu} - \|\mathbf{x}_0\|_1 \|\tilde{\boldsymbol{\beta}}^{(k)}\|_1 \right| - |\tilde{\mu}^{(k)}| \quad i = 1, \ldots n.$$

The norms of $\hat{\mu}^{(k)}$ and $\tilde{\boldsymbol{\beta}}^{(k)}$ are bounded, and hence the right-hand side goes to infinity, as do the residuals for $i \in C$. According to part (i) of Lemma 1, this implies the scale $\hat{\sigma}_{\mathrm{M}}(\tilde{\mathbf{r}}^{(k)})$ tends to infinity as well. The M-estimation equation in the definition of the S-loss can be decomposed to

$$\sum_{i=1}^{m} \rho\left(\frac{\tilde{r}_i^{(k)}}{\hat{\sigma}_{\mathrm{M}}(\tilde{\mathbf{r}}^{(k)})}\right) + \sum_{i=m+1}^{n} \rho\left(\frac{\tilde{r}_i^{(k)}}{\hat{\sigma}_{\mathrm{M}}(\tilde{\mathbf{r}}^{(k)})}\right) = n\delta.$$

Taking the limit for $k \to \infty$, the argument in the ρ function of the second sum tends to zero because the residuals of uncontaminated observations remain bounded, which in turn leads to the second sum converging to 0. The summands in the first term, on the other hand, are all identical and the limit must be

$$\lim_{k\to\infty} \rho\left(\frac{1 - \tilde{\mu}^{(k)}/k^{\nu+1} - \mathbf{x}_0^{\intercal}\tilde{\boldsymbol{\beta}}^{(k)}/k^{\nu}}{\hat{\sigma}_{\mathrm{M}}(\tilde{\mathbf{r}}^{(k)})/k^{\nu+1}}\right) = \frac{n\delta}{m}. \tag{B.3}$$

From assumptions [R1] and [R2] the function $\rho(t)$ is continuous and increasing for $t > 0$ such that $\rho(t) < 1 = \rho(\infty)$. Because $n\delta/m < 1 = \rho(\infty)$ there exists a unique value γ such that

$$\rho\left(\frac{1}{\gamma}\right) = \frac{n\delta}{m}. \tag{B.4}$$

The numerator in the argument in (B.3) tends to 1 and due to (B.4) any converging

subsequence of $\hat{\sigma}_{\text{M}}(\tilde{\boldsymbol{\beta}}^{(k)})/k^{\nu+1}$ must have limit γ. Therefore, the boundedness of $\tilde{\boldsymbol{\theta}}^{(k)}$ implies

$$\lim_{k\to\infty} \frac{1}{k^{2\nu+2}} \mathscr{O}_{\text{S}}(\tilde{\mu}^{(k)}, \tilde{\boldsymbol{\beta}}^{(k)}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}) = \gamma^2. \tag{B.5}$$

Next define an unbounded sequence of parameters as $\mu^{(k)} = 0$ and $\boldsymbol{\beta}^{(k)} = \frac{k^\nu}{2}\mathbf{x}_0$. For this sequence of parameters the residuals are

$$r_i^{(k)} = \begin{cases} \frac{k^{\nu+1}}{2} & i = 1, \ldots, m \\ y_i - \frac{k^\nu}{2}\mathbf{x}_0^\intercal \mathbf{x}_i & i = m+1, \ldots, n \end{cases},$$

which all tend to infinity for $k \to \infty$, implying that $\hat{\sigma}_{\text{M}}(\mathbf{r}^{(k)}) \to \infty$. The decomposition of the M-estimation equation yields

$$\sum_{i=1}^{m} \rho\left(\frac{k^{\nu+1}/2}{\hat{\sigma}_{\text{M}}(\mathbf{r}^{(k)})}\right) + \sum_{i=m+1}^{n} \rho\left(\frac{y_i - \frac{k^\nu}{2}\mathbf{x}_0^\intercal \mathbf{x}_i}{\hat{\sigma}_{\text{M}}(\mathbf{r}^{(k)})}\right) = n\delta.$$

Taking the limit for $k \to \infty$ in all terms, the second sum tends to 0 and, following the same argument as before, the limit of the first sum

$$\lim_{k\to\infty} \frac{1}{k^{2\nu+2}} \mathscr{O}_{\text{S}}(\mu^{(k)}, \boldsymbol{\beta}^{(k)}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}) = \frac{\gamma^2}{4}. \tag{B.6}$$

because the L_1 norm of $\mathbf{x}_0$ is finite.

From the limits (B.5) and (B.6) it follows that there exists a k_0 such that for all $k > k_0$

$$\frac{1}{k^{2\nu+2}} \mathscr{O}_{\text{S}}(\mu^{(k)}, \boldsymbol{\beta}^{(k)}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}) < \frac{1}{k^{2\nu+2}} \mathscr{O}_{\text{S}}(\tilde{\mu}^{(k)}, \tilde{\boldsymbol{\beta}}^{(k)}; \lambda, \alpha, \widetilde{\mathscr{Z}}_m^{(k)}),$$

showing that a bounded $\tilde{\boldsymbol{\theta}}^{(k)}$ can not be a global minimum of the PENSE objective function for the contaminated samples.

□

B.2 Asymptotic Properties of Adaptive PENSE

Below are the proofs of asymptotic properties of adaptive PENSE as presented in Section 4.2. For notational simplicity, I drop the intercept term from the model, i.e., the linear model 2.1 is simplified to

$$\mathcal{Y} = \boldsymbol{\mathcal{X}}^\intercal \boldsymbol{\beta}^0 + \mathcal{U}$$

and the joint distribution G_0 of $(\mathcal{Y}, \boldsymbol{\mathcal{X}})$ is written in terms of the error

$$G_0(u, \mathbf{x}) := G_0(y, \mathbf{x}) = G_0(\mathbf{x}) F_0(y - \mathbf{x}^\intercal \boldsymbol{\beta}^0).$$

All the proofs also hold for the model with an intercept term included. Another notational shortcut in the following proofs is to write the M-scale of the residuals in terms of the regression coefficients, i.e.,

$$\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}) = \hat{\sigma}_{\mathrm{M}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

and accordingly the population version, $\sigma_{\mathrm{M}}(\boldsymbol{\beta})$. For all proofs below, I define $\psi(t) = \rho'(t)$ to denote the first derivative of the ρ function in the definition of the M-scale estimate and hence of the S-loss, as well as the mapping $\varphi \colon \mathrm{R} \to [0; c]$ as

$$\varphi(t) := \psi(t) t.$$

B.2.1 Preliminary Results Concerning the M-Scale Estimator

Before proving asymptotic properties of the adaptive PENSE estimator, several intermediate results concerning the M-scale estimator are required.

Lemma 2. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies (2.2) and $u_i = y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta}^0$. If $\mathbf{v} \in \mathbb{R}^p$ and $s \in (0, \infty)$ positive, then the empirical processes $(P_n \eta_{\mathbf{v},s})_{\mathbf{v},s}$ with*

$$\eta_{\mathbf{v},s}(u, \mathbf{x}) := \varphi\left(\frac{u + \mathbf{x}^\intercal \mathbf{v}}{s}\right)$$

converge uniformly almost sure:

$$\lim_{n \to \infty} \sup_{\substack{\mathbf{v} \in \mathbb{R}^p \\ s \in (0, \infty)}} \left| \frac{1}{n} \sum_{i=1}^{n} \eta_{\mathbf{v},s}(u_i, \mathbf{x}_i) - \mathbb{E}_{G_0}\left[\eta_{\mathbf{v},s}(\mathcal{U}, \boldsymbol{\mathcal{X}})\right] \right| = 0 \quad a.s. \tag{B.7}$$

Proof of Lemma 2. I will show step by step that the space $\mathscr{F} = \{\eta_{\mathbf{v},s} : \mathbf{v} \in \mathbb{R}^p, s \in (0, \infty)\}$ is a bounded Vapnik–Chervonenkis (VC) class of functions and hence Glivenko-Cantelli. The space $\mathscr{F}$ is bounded because $\varphi(t)$ is bounded by assumptions on ρ. Define the mapping

$$g_{\mathbf{v},s} := \begin{cases} \mathbb{R}^{p+1} & \to \mathbb{R} \\ \begin{pmatrix} u \\ \mathbf{x} \end{pmatrix} & \mapsto (u - \mathbf{x}^\intercal \mathbf{v}) s^{-1} \end{cases} .$$

The corresponding function space $\mathscr{G} = \{g_{\mathbf{v},s} : \mathbf{v} \in \mathbb{R}^p, s \in (0,\infty)\}$ is a subset of a finite-dimensional vector space with dimension $\dim(\mathscr{G}) = p+1$. Therefore, $\mathscr{G}$ is VC with VC index $V(\mathscr{G}) \leq p+3$ according to Lemma 2.6.15 in van der Vaart and Wellner (1996). Due to the assumptions on ρ, the function $\varphi(t)$ can be decomposed into

$$\varphi(t) = \max\{\min\{\varphi_1(t), \varphi_2(t)\}, \min\{\varphi_1(-t), \varphi_2(-t)\}\}$$

with $\varphi_{1,2}$ monotone functions. Thus, $\Phi_{1,2} = \{\varphi_{1,2}(g(\cdot)) : g \in \mathscr{G}\}$ and $\Phi_{1,2}^{(-)} = \{\varphi_{1,2}(-g(\cdot)) : g \in \mathscr{G}\}$ are also VC due to Lemma 2.6.18 (iv) and (viii) in van der Vaart and Wellner (1996). Using Lemma 2.6.18 (i) in van der Vaart and Wellner (1996) then leads to $\Phi = \Phi_1 \wedge \Phi_2$ and $\Phi^{(-)} = \Phi_1^{(-)} \wedge \Phi_2^{(-)}$ also being VC. Finally, $\mathscr{F} = \Phi \vee \Phi^{(-)}$ is VC because of Lemma 2.6.18 (ii). Since $\mathscr{F}$ is bounded, Theorem 2.4.3 in van der Vaart and Wellner (1996) concludes the proof. □

Lemma 3. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies (2.2) and $u_i = y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta}^0$. Under assumptions [A1], [A2] and if $\boldsymbol{\beta}_n^* = \boldsymbol{\beta}^0 + \mathbf{v}_n$ with $\lim_{n\to\infty} \|\mathbf{v}_n\| = 0$ a.s., then we have*

(a) almost sure convergence of the estimated M-scale to the population M-scale of the error distribution

$$\lim_{n\to\infty} \hat{\sigma}_M(\boldsymbol{\beta}_n^*) \xrightarrow{a.s.} \sigma_M(\boldsymbol{\beta}^0)$$

(b) and almost sure convergence of

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \varphi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n}{\hat{\sigma}_M(\boldsymbol{\beta}_n^*)}\right) = \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_M(\boldsymbol{\beta}^0)}\right)\right] \quad a.s.$$

Proof of Lemma 3. The first result (a) is a direct consequence of the conditions of the lemma ($u - \mathbf{x}^\intercal \mathbf{v}_n \to u$ a.s.) and Theorem 3.1 in Yohai (1987).

For part (b), it is know from Lemma 2 the empirical process converges uniformly almost sure. Since $\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) > 0$, the continuous mapping theorem gives $\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)} \to \frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}$ almost surely. Finally, due to the continuity and boundedness of φ:

$$\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_n}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)}\right)\right] \xrightarrow[n\to\infty]{\text{a.s.}} \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right] \tag{B.8}$$

which concludes the proof. □

Lemma 4. *Let $(y_i, \mathbf{x}_i^\intercal)$, $i = 1, \ldots, n$, be i.i.d. observations with distribution G_0 which satisfies (2.2) and $u_i = y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta}^0$. Under regularity conditions [A1]–[A3] and if $\mathbf{v} \in K \subset \mathbb{R}^p$ with K compact and $\boldsymbol{\beta}_n^* = \boldsymbol{\beta}^0 + \mathbf{v}/\sqrt{n}$, then*

(a) the M-scale estimate converges uniformly almost sure

$$\sup_{\mathbf{v} \in K} \left| \hat{\sigma}_M(\boldsymbol{\beta}_n^*) - \sigma_M(\boldsymbol{\beta}^0) \right| \xrightarrow{a.s.} 0, \tag{B.9}$$

(b) for every $\epsilon > 0$ with $\epsilon < \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_M(\boldsymbol{\beta}^0)}\right)\right]$ the uniform bound over $\mathbf{v} \in K$

$$\sup_{\mathbf{v} \in K} \left| \frac{\hat{\sigma}_M(\boldsymbol{\beta}_n^*)}{\frac{1}{n}\sum_{i=1}^n \varphi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}/\sqrt{n}}{\hat{\sigma}_M(\boldsymbol{\beta}_n^*)}\right)} \right| < \frac{\epsilon + \sigma_M(\boldsymbol{\beta}^0)}{\mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_M(\boldsymbol{\beta}^0)}\right)\right] - \epsilon} \tag{B.10}$$

holds with arbitrarily high probability if n is sufficiently large.

Proof of Lemma 4. The proof for (B.9) relies on Lemma 4.5 from Yohai and Zamar (1986) which states that under the same conditions as for this lemma, the following holds:

$$\sup_{\mathbf{v} \in K} |\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}_n^*)| \xrightarrow{\text{a.s.}} 0.$$

Therefore, the missing step is to show that $\sup_{\mathbf{v} \in K} |\sigma_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| \to 0$ almost surely as $n \to \infty$. This is done by contradiction.

Assume there exists a subsequence $(n_k)_{k>0}$ such that for all k, $\sup_{\mathbf{v} \in K} |\sigma_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| > \epsilon > 0$. Since $\mathbf{v} \in K$ with K a compact set, for every sequence $\mathbf{v}_n$ there exists a subsequence $(\mathbf{v}_{n_k})_k$ such that $|\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k}) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| > \epsilon$ for all $n_k > N_\epsilon$. Therefore, either one of the following holds: (i) $\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k}) > \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon$ or (ii) $\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k}) < \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) - \epsilon$. In the first case (i) it is know that

$$\rho\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_{n_k}/\sqrt{n}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k})}\right) < \rho\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_{n_k}/\sqrt{n}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon}\right) \to \rho\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon}\right).$$

Due to the boundedness of ρ, the dominated convergence theorem gives

$$\mathbb{E}_{G_0}\left[\rho\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_{n_k}/\sqrt{n}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k})}\right)\right] < \mathbb{E}_{G_0}\left[\rho\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_{n_k}/\sqrt{n}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon}\right)\right] \to \mathbb{E}_{G_0}\left[\rho\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon}\right)\right] < \delta$$

which contradicts the definition of $\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k})$. In case (ii) similar steps yield

$$\mathbb{E}_{G_0}\left[\rho\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}_{n_k}/\sqrt{n}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k})}\right)\right] > \delta$$

for all $n_k > N$ with N large enough. Therefore, the assumption $\sup_{\mathbf{v}\in K} |\sigma_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| > \epsilon > 0$ can not be valid and hence $\sup_{\mathbf{v}\in K} |\sigma_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| \to 0$. This concludes the proof of (B.9).

Before proving (B.10), note that ϵ is well defined because $\mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right] > 0$ as per Lemma 6 in Smucler (2019). To prove (B.10), I first bound the denominator uniformly over $\mathbf{v} \in K$. From Lemma 2 it is known that the empirical processes converge almost surely, uniformly over $\mathbf{v} \in K$ and $s > 0$. As a next step, I show the deterministic uniform convergence of

$$\sup_{\substack{\mathbf{v}\in K \\ s\in[\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)-\epsilon_1, \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)+\epsilon_1]}} \left|\mathbb{E}_{G_0}\left[f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)\right] - \mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U}}{s}\right)\right]\right| \to 0, \tag{B.11}$$

where $f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)$ is defined as

$$f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s) := \varphi\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal \mathbf{v}/\sqrt{n}}{s}\right).$$

The functions $f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)$ are bounded and converge pointwise to $\varphi\left(\frac{\mathcal{U}}{s}\right)$, entailing pointwise convergence of $\mathbb{E}_{G_0}\left[f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)\right] \to \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{s}\right)\right]$ as $n \to \infty$ by the dominated convergence theorem. Because ρ has bounded second derivative, the derivative of $f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)$ with respect to $\mathbf{v} \in K$ and $s \in [\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) - \epsilon_1, \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0) + \epsilon_1]$ is also bounded, meaning $f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, s)$ is equicontinuous on this domain. Pointwise convergence together with the equicontinuity make the Arzelà-Ascoli theorem applicable and hence conclude that (B.11) holds.

From (B.9) it follows that for any $\delta_2 > 0$ there is a N_{δ_2} such that for all $\mathbf{v} \in K$ and all $n > N_{\delta_2}$, $\mathbb{P}\left(|\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*) - \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)| \leq \epsilon_1\right) > 1 - \delta_2$. Combined with (B.11) this yields that for every $\delta_2 > 0$ and $\epsilon_2 > 0$ there is an N_{δ_2,ϵ_2} such that for all $n > N_{\delta_2,\epsilon_2}$ and every $\mathbf{v} \in K$

$$\left|\mathbb{E}_{G_0}\left[f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, \hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*))\right] - \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)}\right)\right]\right| < \epsilon_2$$

with probability greater than $1 - \delta_2$. Since both expected values are positive this can also

be written as

$$\mathbb{E}_{G_0}\left[f_n(\mathcal{U}, \boldsymbol{\mathcal{X}}, \mathbf{v}, \hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*))\right] > \mathbb{E}_{F_0}\left[\varphi\left(\frac{u}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)}\right)\right] - \epsilon_2. \tag{B.12}$$

The final piece for the denominator to be bounded is to show that

$$\sup_{\mathbf{v}\in K}\left|\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)}\right)\right] - \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]\right| \xrightarrow[n\to\infty]{\text{a.s.}} 0. \tag{B.13}$$

Set $\Omega_1 = \{\omega : \hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*;\omega) \to \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)\}$ which has $\mathbb{P}(\Omega_1) = 1$ due to the first part of this lemma. Similarly, set $\Omega_2 = \{\omega :$ equation (B.13) holds$\}$. Assume now that $\mathbb{P}(\Omega_1 \cap \Omega_2^\complement) > 0$. This assumption entails that there exists an $\omega' \in \Omega_1 \cap \Omega_2^\complement$, an $\epsilon_3 > 0$ and a subsequence $(n_k)_{k>0}$ such that

$$\lim_{k\to\infty}\left|\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}^0 + \frac{\mathbf{v}_{n_k}}{\sqrt{n_k}};\omega')}\right)\right] - \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]\right| > \epsilon_3. \tag{B.14}$$

However, since $\mathbf{v}_{n_k}$ is in the compact set K, the sequence $\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k}$ converges to $\boldsymbol{\beta}^0$ as $n \to \infty$. Additionally, φ is bounded and together with the dominated convergence theorem this leads to

$$\lim_{k\to\infty}\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k};\omega')}\right)\right] = \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]$$

and in turn to

$$\lim_{k\to\infty}\left|\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_{n_k}/\sqrt{n_k};\omega')}\right)\right] - \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]\right| = 0$$

contradicting the claim in (B.14). Therefore, $\mathbb{P}(\Omega_1 \cap \Omega_2^\complement) = 0$, proving (B.13). Combining (B.12) and (B.13) leads to the conclusion that with arbitrarily high probability for large enough n

$$\left|\mathbb{E}_{G_0}\left[\varphi\left(\frac{\mathcal{U} - \boldsymbol{\mathcal{X}}^\intercal\mathbf{v}/\sqrt{n}}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)}\right)\right]\right| > -\epsilon_4 + \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right] \tag{B.15}$$

for every $\mathbf{v} \in K$.

From the first part of this lemma, $\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*) \xrightarrow{\text{a.s.}} \sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)$, and due to (B.15), for every $\delta > 0$ and every $0 < \epsilon < \mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]$ there exists an $N_{\delta,\epsilon}$ such that for all $\mathbf{v} \in K$ and $n \geq N_{\delta,\epsilon}$ equation (B.10) holds. □

B.2.2 Root-n Consistency

Proof of Theorem 3. To ease the notation for the proof, the hyper-parameters are dropped from the objective function $\mathcal{L}_{\text{AS}}$ and the adaptive elastic net penalty is simply denoted by $\Phi(\boldsymbol{\beta}) = \Phi_{\text{AN}}(\boldsymbol{\beta}; \lambda_{\text{AS}}, \alpha_{\text{AS}}, \zeta, \boldsymbol{\omega})$. Also, $\boldsymbol{\gamma}(t) := \boldsymbol{\beta}^0 + t(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)$ denotes the convex combination of the true parameter $\boldsymbol{\beta}^0$ and the adaptive PENSE estimator $\hat{\boldsymbol{\beta}}$. It is important to remember the penalty loadings are derived from a preliminary PENSE estimator, $\tilde{\boldsymbol{\beta}}$, $\boldsymbol{\omega} = (1/\tilde{\beta}_1, \ldots, 1/\tilde{\beta}_p)^\intercal$.

The first step in the proof is a Taylor expansion of the objective function around the true parameter $\boldsymbol{\beta}^0$:

$$\begin{aligned}\hat{\sigma}^2_{\text{M}}(\hat{\boldsymbol{\beta}}) + \Phi(\hat{\boldsymbol{\beta}}) =& \hat{\sigma}^2_{\text{M}}(\boldsymbol{\beta}^0) + \Phi(\boldsymbol{\beta}^0) + (\Phi(\hat{\boldsymbol{\beta}}) - \Phi(\boldsymbol{\beta}^0)) \\ &- 2 \underbrace{\frac{1}{\frac{1}{n}\sum_{i=1}^n \varphi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right)}}_{=:A_n} \underbrace{\frac{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}{n} \sum_{i=1}^n \psi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i^\intercal \mathbf{v}_n}_{=:Z_n}\end{aligned}$$

where $\mathbf{v}_n = \tau(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)$ and $\boldsymbol{\beta}_n^* = \boldsymbol{\beta}^0 + \mathbf{v}_n$ for a $0 < \tau < 1$. Due to the strong consistency of $\hat{\boldsymbol{\beta}}$ from Proposition 2, $\mathbf{v}_n \to 0$ a.s. and hence from Lemma 3 and the continuous mapping theorem it is know that $A_n \xrightarrow{\text{a.s.}} \frac{1}{\mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\text{M}}(\boldsymbol{\beta}^0)}\right)\right]} =: A > 0$ as well as $\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) \xrightarrow{\text{a.s.}} \sigma_{\text{M}}(\boldsymbol{\beta}^0)$. The term Z_n is handled by a Taylor expansion of $\psi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right)$ around u_i to get

$$\begin{aligned}Z_n &= \hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) \left(\frac{1}{n} \sum_{i=1}^n \psi\left(\frac{u_i}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i^\intercal \mathbf{v}_n - \frac{1}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) n} \sum_{i=1}^n \psi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n^*}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i^\intercal \mathbf{v}_n \mathbf{x}_i^\intercal \mathbf{v}_n \right) \\ &= \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal}{\sqrt{n}} \left[\tau \hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) \frac{1}{\sqrt{n}} \sum_{i=1}^n \psi\left(\frac{u_i}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i \right] \\ &\quad - \tau^2 (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal \left[\frac{1}{n} \sum_{i=1}^n \psi'\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n^*}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i \mathbf{x}_i^\intercal \right] (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)\end{aligned}$$

for some $\mathbf{v}_n^* = \tau^* \mathbf{v}_n$ with $\tau^* \in (0, 1)$.

The rest of the proof follows closely the proof of Proposition 2 in Smucler and Yohai (2017). More specifically, noting that $\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) \xrightarrow{\text{a.s.}} \sigma_{\text{M}}(\boldsymbol{\beta}^0)$, the results in Smucler and Yohai (2017) (which are derived from results in Yohai (1985)) state that

$$B_n := \|\boldsymbol{\xi}_n\| = O_p(1) \quad \text{with} \quad \boldsymbol{\xi}_n = \tau \hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*) \frac{1}{\sqrt{n}} \sum_{i=1}^n \psi\left(\frac{u_i}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}_n^*)}\right) \mathbf{x}_i$$

and hence with arbitrarily high probability for n sufficiently large there is a B such that

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal}{\sqrt{n}} \boldsymbol{\xi}_n \leq \frac{1}{\sqrt{n}} \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| \|\boldsymbol{\xi}_n\| \leq \frac{B}{\sqrt{n}} \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|. \tag{B.16}$$

Similarly, the results in Smucler and Yohai (2017) can be used to show

$$C_n := \tau^2 (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0)^\intercal \left[\frac{1}{n} \sum_{i=1}^{n} \psi' \left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n^*}{\hat{\sigma}_{\mathrm{M}}(\boldsymbol{\beta}_n^*)} \right) \mathbf{x}_i \mathbf{x}_i^\intercal \right] (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0) \geq \widetilde{C}_n \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|^2 \tag{B.17}$$

with $\widetilde{C}_n \xrightarrow{\text{a.s.}} C > 0$.

Next is the difference in the penalty terms $D_n := \Phi(\hat{\boldsymbol{\beta}}) - \Phi(\boldsymbol{\beta}^0)$, which can be reduced to the truly non-zero coefficients:

$$\begin{aligned} D_n =& \lambda_{\mathrm{AS},n} \sum_{j=1}^{p} \left(\frac{1-\alpha}{2} \left((\hat{\beta}_j)^2 - (\beta_j^0)^2 \right) + \alpha \frac{|\hat{\beta}_j| - |\beta_j^0|}{|\tilde{\beta}_j|^\zeta} \right) \\ \geq& \lambda_{\mathrm{AS},n} \sum_{j=1}^{s} \left(\frac{1-\alpha}{2} \left((\hat{\beta}_j)^2 - (\beta_j^0)^2 \right) + \alpha \frac{|\hat{\beta}_j| - |\beta_j^0|}{|\tilde{\beta}_j|^\zeta} \right). \end{aligned}$$

Observing that $\hat{\boldsymbol{\beta}}$ is a strongly consistent estimator, $|\hat{\beta}_j - \beta_j^0| < \epsilon_j < |\beta_j^0|$ for all $j = 1, \ldots, s$ and any $\epsilon_j \in (0, |\beta_j^0|)$ with arbitrarily high probability for sufficiently large n. This entails that, for all $0 \leq t \leq 1$ and $j = 1, \ldots, s$, the sign of the convex combination $\operatorname{sgn}(\gamma_j(t)) = \operatorname{sgn}(\beta_j^0) \neq 0$ and thus $|\gamma_j(t)|$ is differentiable. This allows application of the mean value theorem on the quadratic and the absolute term in D_n to yield

$$D_n \geq \lambda_{\mathrm{AS},n} \sum_{j=1}^{s} \left(\frac{1-\alpha}{4} \gamma_j(\tau_j) + \alpha \frac{\operatorname{sgn}(\beta_j^0)}{|\tilde{\beta}_j|^\zeta} \right) (\hat{\beta}_j - \beta_j^0)$$

for some $\tau_j \in (0,1)$, $j = 1, \ldots, s$, with arbitrarily high probability for large enough n. Because both $\tilde{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}$ are strongly consistent for $\boldsymbol{\beta}^0$ and $\lambda_{\mathrm{AS},n} = O(1/\sqrt{n})$, there exists a constant D such that with arbitrarily high probability

$$D_n \geq -\frac{D}{\sqrt{n}} \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| \tag{B.18}$$

for sufficiently large n.

Since $\hat{\boldsymbol{\beta}}$ minimizes the adaptive PENSE objective function $\mathcal{L}_{\text{AS}}$,

$$0 \geq \mathcal{L}_{\text{AS}}(\hat{\boldsymbol{\beta}}) - \mathcal{L}_{\text{AS}}(\boldsymbol{\beta}^0) = \hat{\sigma}^2_{\text{M}}(\hat{\boldsymbol{\beta}}) + \Phi(\hat{\boldsymbol{\beta}}) - \hat{\sigma}^2_{\text{M}}(\boldsymbol{\beta}^0) - \Phi(\boldsymbol{\beta}^0) = D_n - 2A_nZ_n.$$

With the bounds derived in (B.16), (B.17), and (B.18) this in turn yields

$$\begin{aligned}
0 \geq & D_n - 2A_nZ_n = D_n - 2A_nB_n + 2A_nC_n \\
\geq & -\frac{D}{\sqrt{n}}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| - 2A\frac{B}{\sqrt{n}}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| + 2AC\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|^2 \\
= & \frac{1}{\sqrt{n}}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| \left(-D - 2AB + 2AC\sqrt{n}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|\right)
\end{aligned}$$

with arbitrarily high probability for large enough n. Rearranging the terms leads to the inequality

$$\sqrt{n}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| \leq \frac{2AB + D}{2AC}.$$

□

B.2.3 Variable Selection Consistency

Proof of Theorem 4. To ease notation in the following, I denote the coordinate-wise adaptive EN penalty function by

$$\phi(\beta; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega) = \lambda_{\text{AS},n}\left(\frac{1 - \alpha_{\text{AS}}}{2}\beta^2 + \alpha_{\text{AS}}|\beta||\omega|^\zeta\right)$$

such that $\lambda_{\text{AS},n}\Phi_{\text{AN}}(\boldsymbol{\beta}; \alpha_{\text{AS}}, \zeta, \boldsymbol{\omega}) = \sum_{j=1}^p \phi(\beta_j; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega_j)$. I follow the proof in Smucler and Yohai (2017) and define the function

$$\begin{aligned}
V_n(\mathbf{v}_1, \mathbf{v}_2) := & \hat{\sigma}^2_{\text{M}}(\boldsymbol{\beta}^0_{\text{I}} + \mathbf{v}_1/\sqrt{n}, \boldsymbol{\beta}^0_{\text{II}} + \mathbf{v}_2/\sqrt{n}) + \\
& \sum_{j=1}^{s} \phi(\beta^0_j + v_{1,j}/\sqrt{n}; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega_j) + \\
& \sum_{j=s+1}^{p} \phi(\beta^0_j + v_{2,j-s}/\sqrt{n}; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega_j).
\end{aligned}$$

From Theorem 3 follows with arbitrarily high probability, $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\| \leq C/\sqrt{n}$ for sufficiently large n. Therefore, with arbitrarily high probability $V_n(\mathbf{v}_1, \mathbf{v}_2)$ attains its minimum on the compact set $\left\{(\mathbf{v}_1, \mathbf{v}_2) : \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2 \leq C^2\right\}$ at $\hat{\boldsymbol{\beta}}$. The goal is to show that for any

$\|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2 \leq C^2$ with $\|\mathbf{v}_2\| > 0$ and with arbitrarily high probability, $V_n(\mathbf{v}_1, \mathbf{v}_2) - V_n(\mathbf{v}_1, \mathbf{0}_{p-s}) > 0$ for sufficiently large n.

Taking the difference while observing that $\boldsymbol{\beta}_{\text{II}}^0 = \mathbf{0}_{p-s}$ gives

$$V_n(\mathbf{v}_1, \mathbf{v}_2) - V_n(\mathbf{v}_1, \mathbf{0}_{p-s}) = \big(\hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}_{\text{I}}^0 + \mathbf{v}_1/\sqrt{n}, \mathbf{v}_2/\sqrt{n}) - \hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}_{\text{I}}^0 + \mathbf{v}_1/\sqrt{n}, \mathbf{0}_{p-s})\big) + \sum_{j=s+1}^{p} \phi(v_{2,j-s}/\sqrt{n}; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega_j).$$

The first term can be bounded by defining $\mathbf{v}_n(t) := (\mathbf{v}_1^\intercal, t\mathbf{v}_2^\intercal)^\intercal/\sqrt{n}$ and applying the mean value theorem gives some $\tau \in (0, 1)$ such that

$$\begin{aligned}
&\hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}^0 + \mathbf{v}_n(1)) - \hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}^0 + \mathbf{v}_n(0)) = \\
&\frac{2}{\sqrt{n}}\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_n(\tau))(\mathbf{0}_s^\intercal, \mathbf{v}_2^\intercal)\boldsymbol{\nabla}_{\boldsymbol{\beta}}\hat{\sigma}_{\text{M}}(\boldsymbol{\beta})\big|_{\boldsymbol{\beta}^0+\mathbf{v}_n(\tau)} = \\
&-\frac{2}{\sqrt{n}} \underbrace{\frac{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_n(\tau))}{\frac{1}{n}\sum_{i=1}^n \varphi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n(\tau)}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}^0+\mathbf{v}_n(\tau))}\right)}}_{=:A_n} \underbrace{(\mathbf{0}_s^\intercal, \mathbf{v}_2^\intercal)\frac{1}{n}\sum_{i=1}^n \psi\left(\frac{u_i - \mathbf{x}_i^\intercal \mathbf{v}_n(\tau)}{\hat{\sigma}_{\text{M}}(\boldsymbol{\beta}^0 + \mathbf{v}_n(\tau))}\right)\mathbf{x}_i}_{=:B_n}.
\end{aligned}$$

By Lemma 4 the term A_n is uniformly bounded in probability, hence $|A_n| < A$ with arbitrarily high probability for large enough n. Furthermore, $|B_n| \leq \|\psi\|_\infty \|\mathbf{v}_2\| \left\|\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right\|$ and due to the law of large numbers there is a constant B such that the upper bound for $|B_n|$ is

$$|B_n| \leq \|\psi\|_\infty \|\mathbf{v}_2\| (\|\mathbb{E}_{H_0}[\boldsymbol{\mathcal{X}}]\| + \epsilon) < \|\mathbf{v}_2\| B$$

with arbitrarily high probability for sufficiently large n. Together, the bounds for A_n and B_n give

$$\hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}^0 + \mathbf{v}_n(1)) - \hat{\sigma}_{\text{M}}^2(\boldsymbol{\beta}^0 + \mathbf{v}_n(0)) \geq -\frac{\|\mathbf{v}_2\|}{\sqrt{n}} 2AB. \tag{B.19}$$

The next step is to ensure that the penalty term grows large enough to make the difference $V_n(\mathbf{v}_1, \mathbf{v}_2) - V_n(\mathbf{v}_1, \mathbf{0}_{p-s})$ positive. Indeed, the assumption $\alpha_{\text{AS}} > 0$ and using a PENSE estimator for the penalty loadings, $\omega_j = 1/|\tilde{\beta}_j|$ leads to

$$\begin{aligned}
\sum_{j=s+1}^{p} \phi(v_{2,j-s}/\sqrt{n}; \lambda_{\text{AS},n}, \alpha_{\text{AS}}, \zeta, \omega_j) \geq &\alpha_{\text{AS}}\lambda_{\text{AS},n} \sum_{j=s+1}^{p} \frac{|v_{2,j-s}|}{\sqrt{n}|\tilde{\beta}_j|^\zeta} \\
= &\alpha_{\text{AS}}\lambda_{\text{AS},n} n^{(\zeta-1)/2} \sum_{j=s+1}^{p} \frac{|v_{2,j-s}|}{|\sqrt{n}\tilde{\beta}_j|^\zeta}.
\end{aligned}$$

The root-n consistency of $\tilde{\boldsymbol{\beta}}$ established in Theorem 1 gives $|\sqrt{n}\tilde{\beta}_j| < M$ with arbitrarily high probability for large enough n. Therefore,

$$\begin{aligned}\alpha_{\text{AS}}\lambda_{\text{AS},n}n^{(\zeta-1)/2}n^{(\zeta-1)/2}\sum_{j=s+1}^{p}\frac{|v_{2,j-s}|}{|\sqrt{n}\tilde{\beta}_j|^{\zeta}} &> \alpha_{\text{AS}}\lambda_{\text{AS},n}n^{(\zeta-1)/2}n^{(\zeta-1)/2}\sum_{j=s+1}^{p}\frac{|v_{2,j-s}|}{M^{\zeta}} \\ &= \alpha_{\text{AS}}\lambda_{\text{AS},n}n^{(\zeta-1)/2}n^{(\zeta-1)/2}\frac{\|v_2\|_1}{M^{\zeta}} \\ &\geq \frac{\|v_2\|}{\sqrt{n}M^{\zeta}}\alpha_{\text{AS}}\lambda_{\text{AS},n}n^{(\zeta-1)/2}n^{\zeta/2}.\end{aligned} \tag{B.20}$$

Combining (B.19) and (B.20) yields

$$V_n(\mathbf{v}_1,\mathbf{v}_2) - V_n(\mathbf{v}_1,\mathbf{0}_{p-s}) > \frac{\|\mathbf{v}_2\|}{\sqrt{n}}\left(-2AB + \frac{\alpha_{\text{AS}}\lambda_{\text{AS},n}n^{\zeta/2}}{M^{\zeta}}\right) \tag{B.21}$$

uniformly over $\mathbf{v}_1$ and $\mathbf{v}_2$ with arbitrarily high probability for sufficiently large n. By assumption $\alpha_{\text{AS}}\lambda_{\text{AS},n}n^{\zeta/2} \to \infty$ and hence the right-hand side in (B.21) will eventually be positive, concluding the proof. □

B.2.4 Asymptotic Normal Distribution

Proof of Theorem 5. For this proof I denote the values of the active predictors and the active predictors in the i-th observation by $\mathbf{X}_{\text{I}}$ and $\mathbf{x}_{i,\text{I}}$, respectively. Because $\hat{\boldsymbol{\beta}}$ is strongly consistent for $\boldsymbol{\beta}^0$, the coefficient values for the truly active predictors are almost surely bounded away from zero if n is large enough. This entails that the partial derivatives of the penalty function exist for the truly active predictors and the gradient at the estimate $\hat{\boldsymbol{\beta}}$ is

$$\mathbf{0}_s = \boldsymbol{\nabla}_{\boldsymbol{\beta}_{\text{I}}}\mathcal{L}_{\text{AS}}(\hat{\boldsymbol{\beta}}) = -2\frac{\hat{\sigma}_{\text{M}}(\hat{\boldsymbol{\beta}})}{A_n}\frac{1}{n}\sum_{i=1}^{n}\psi\left(\frac{y_i - \mathbf{x}_i^{\intercal}\hat{\boldsymbol{\beta}}}{\hat{\sigma}_{\text{M}}(\hat{\boldsymbol{\beta}})}\right)\mathbf{x}_{i,\text{I}} + \boldsymbol{\nabla}_{\boldsymbol{\beta}_{\text{I}}}\Phi_{\text{AN}}(\hat{\boldsymbol{\beta}};\lambda_{\text{AS},n},\alpha_{\text{AS}},\zeta,\boldsymbol{\omega}) \tag{B.22}$$

with $A_n = \frac{1}{n}\sum_{i=1}^{n}\varphi\left(\frac{y_i-\mathbf{x}_i^{\intercal}\hat{\boldsymbol{\beta}}}{\hat{\sigma}_{\text{M}}(\hat{\boldsymbol{\beta}})}\right)$. The truly active coefficients can be separated from the truly inactive coefficients by noting that $\psi\left(\frac{y_i-\mathbf{x}_i^{\intercal}\hat{\boldsymbol{\beta}}}{\hat{\sigma}_{\text{M}}(\hat{\boldsymbol{\beta}})}\right) = \psi\left(\frac{y_i-\mathbf{x}_{i,\text{I}}^{\intercal}\hat{\boldsymbol{\beta}}_{\text{I}}}{\hat{\sigma}_{\text{M}}(\hat{\boldsymbol{\beta}})}\right) + o_i$ for some o_i which vanishes in probability, $\mathbb{P}(o_i = 0) \to 1$, because of Theorem 4 and because ψ is continuous.

Equation (B.22) can now be written as

$$\begin{aligned}\mathbf{0}_s =& -2\frac{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}{A_n}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi\left(\frac{y_i-\mathbf{x}_{i,\mathrm{I}}^{\intercal}\hat{\boldsymbol{\beta}}_{\mathrm{I}}}{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}\right)\mathbf{x}_{i,\mathrm{I}}\\ &-2\frac{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}{A_n}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}o_i\mathbf{x}_{i,\mathrm{I}}\\ &+\sqrt{n}\boldsymbol{\nabla}_{\boldsymbol{\beta}_{\mathrm{I}}}\Phi_{\mathrm{AN}}(\hat{\boldsymbol{\beta}};\lambda_{\mathrm{AS},n},\alpha_{\mathrm{AS}},\zeta,\boldsymbol{\omega})\end{aligned}$$

and using the mean value theorem there are $\tau_i\in[0,1]$ and hence a matrix

$$\mathbf{W}_n=\frac{1}{n}\sum_{i=1}^{n}\psi'\left(\frac{u_i-\tau_i\mathbf{x}_{i,\mathrm{I}}^{\intercal}\left(\hat{\boldsymbol{\beta}}_{\mathrm{I}}-\boldsymbol{\beta}_{\mathrm{I}}^{0}\right)}{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}\right)\mathbf{x}_{i,\mathrm{I}}\mathbf{x}_{i,\mathrm{I}}^{\intercal}$$

such that the equation can be further rewritten to

$$\begin{aligned}\mathbf{0}_s =& -2\frac{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}{A_n}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi\left(\frac{y_i-\mathbf{x}_{i,\mathrm{I}}^{\intercal}\boldsymbol{\beta}_{\mathrm{I}}^{0}}{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}\right)\mathbf{x}_{i,\mathrm{I}}\\ &+2\frac{1}{A_n}\mathbf{W}_n\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{\mathrm{I}}-\boldsymbol{\beta}_{\mathrm{I}}^{0}\right)\\ &-2\frac{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}{A_n}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}o_i\mathbf{x}_{i,\mathrm{I}}\\ &+\sqrt{n}\lambda_{\mathrm{AS},n}\boldsymbol{\nabla}_{\boldsymbol{\beta}_{\mathrm{I}}}\Phi_{\mathrm{AN}}(\hat{\boldsymbol{\beta}};\alpha_{\mathrm{AS}},\zeta,\boldsymbol{\omega}).\end{aligned}$$

Separating the term $\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{\mathrm{I}}^{*}-\boldsymbol{\beta}_{\mathrm{I}}^{0}\right)$ then gives

$$\begin{aligned}\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{\mathrm{I}}^{*}-\boldsymbol{\beta}_{\mathrm{I}}^{0}\right) =& \hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})\mathbf{W}_n^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi\left(\frac{y_i-\mathbf{x}_{i,\mathrm{I}}^{\intercal}\boldsymbol{\beta}_{\mathrm{I}}^{0}}{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})}\right)\mathbf{x}_{i,\mathrm{I}}\\ &+\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})\mathbf{W}_n^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}o_i\mathbf{x}_{i,\mathrm{I}}\\ &+\sqrt{n}\lambda_{\mathrm{AS},n}\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})A_n\mathbf{W}_n^{-1}\boldsymbol{\nabla}_{\boldsymbol{\beta}_{\mathrm{I}}}\Phi_{\mathrm{AN}}(\hat{\boldsymbol{\beta}};\alpha_{\mathrm{AS}},\zeta,\boldsymbol{\omega}).\end{aligned}\tag{B.23}$$

The strong consistency of $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}^0$ and Lemma 3 lead to $\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})\xrightarrow{a.s.}\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)$ and $A_n\xrightarrow{a.s.}\mathbb{E}_{F_0}\left[\varphi\left(\frac{\mathcal{U}}{\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)}\right)\right]<\infty$. Also, because of $\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})\xrightarrow{a.s.}\sigma_{\mathrm{M}}(\boldsymbol{\beta}^0)$, Lemma 4.2 in Yohai (1985), and the law of large numbers

$$\mathbf{W}_n\xrightarrow{a.s.}b(\rho,F_0)\boldsymbol{\Sigma}_{\mathrm{I}}.$$

Combined with the assumption that $\sqrt{n}\lambda_{\mathrm{AS},n} \to 0$ this leads to the last two lines in (B.23) converging to $\mathbf{0}_s$ in probability. Finally by Lemma 5.1 in Yohai (1985) and the CLT

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi \left(\frac{y_i - \mathbf{x}_{i,\mathrm{I}}^{\intercal} \boldsymbol{\beta}_{\mathrm{I}}^{0}}{\hat{\sigma}_{\mathrm{M}}(\hat{\boldsymbol{\beta}})} \right) \mathbf{x}_{i,\mathrm{I}} \xrightarrow{d} N_s \left(\mathbf{0}_s, a(\rho, F_0) \boldsymbol{\Sigma}_{\mathrm{I}} \right)$$

which, after applying Slutsky's Theorem, completes the proof. □

Appendix C

Additional Results from Numerical Experiments

C.1 Elastic Net S-Estimators

Below are complete results from the numerical experiments detailed in Section 3.6 including additional estimators, error distributions, and sample sizes. Unregularized MM- and S-estimates are computed only for scenarios where $p < \lfloor (1-\delta)n \rfloor - 1$. The breakdown point of all robust estimators is set to $\delta = 0.33$. Oracle MM- and S-estimators are computed using only the truly active predictors.

C.1.1 Prediction Performance

Prediction performance is measured in terms of the relative scale of the prediction error, as detailed in Section 3.6.3. Figures C.1 and C.2 show results for very sparse scenarios ($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

C.1.2 Variable Selection Performance

Variable selection performance is summarized by the sensitivity (i.e., the proportion of truly active predictors detected as such) and specificity (i.e., the proportion of truly inactive predictors detected as such). The summary figures show sensitivity and specificity in a single plot for regularized estimators only. Sensitivity extends upwards, specificity extends downwards. Methods perform well in terms of variable selection if the two points are at the top and bottom ends of the plot. Figures C.3 and C.4 show results for very sparse scenarios

($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

C.1.3 Estimation Accuracy

The focus of this work is on prediction performance and variable selection of estimators in the linear regression model. To underline consistency of the estimator, however, estimation accuracy is also of interest. Estimation accuracy is assessed by the L_2 estimation error,

$$\text{RMSE}(\hat{\boldsymbol{\beta}}) = \sqrt{\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\right\|_2^2 + (\hat{\mu} - \mu^0)^2}.$$

As detailed in Section 2.1, the L_2 estimation error is similar to the RMSPE, but possible dependence between predictors is ignored. The L_2 estimation error captures both the bias and variance of the estimator. The smaller the L_2 estimation error, the more accurate the estimation. Figures C.5 and C.6 show results for very sparse scenarios ($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

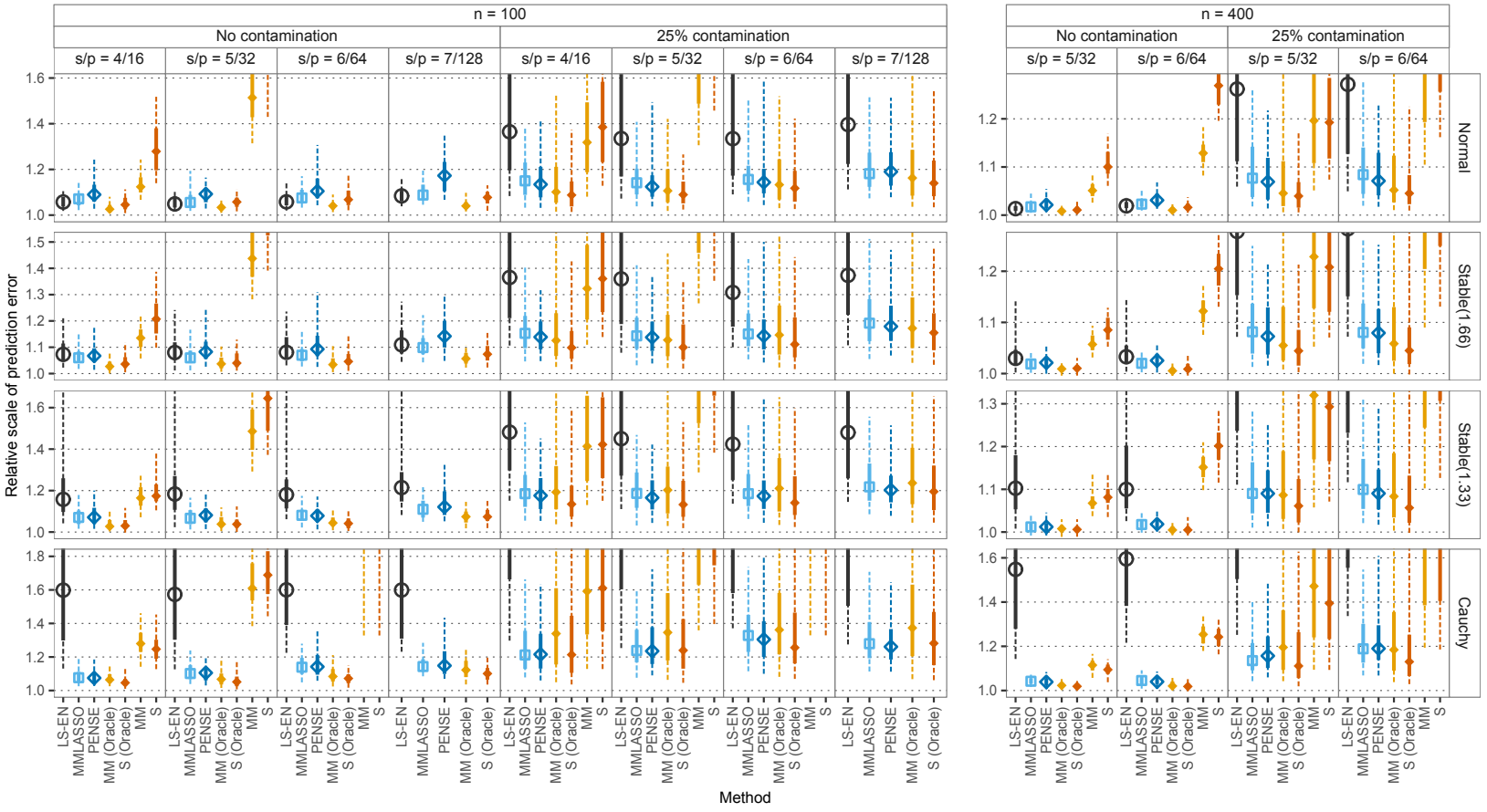


Figure C.1: Prediction performance of estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

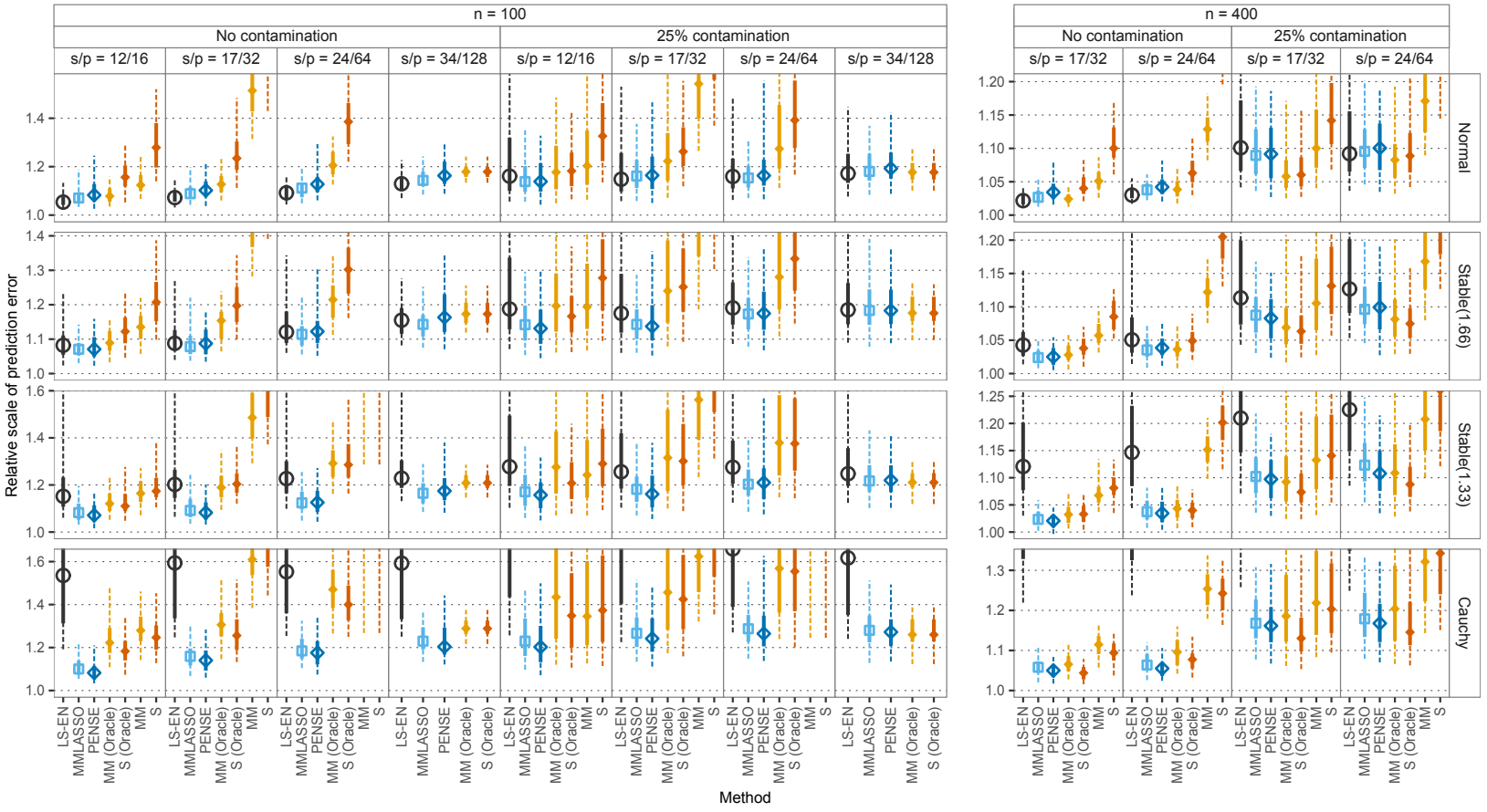


Figure C.2: Prediction performance of estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

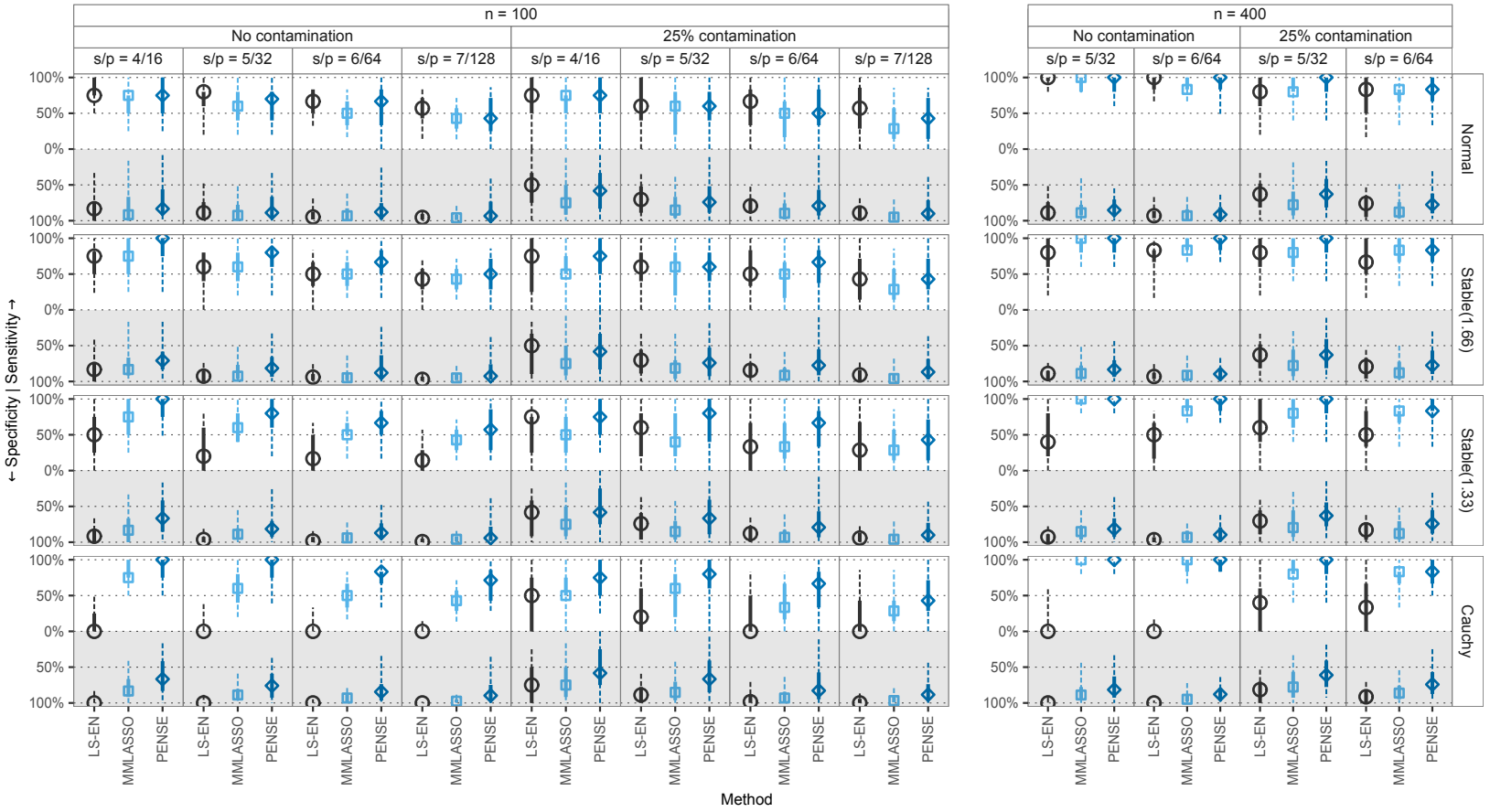


Figure C.3: Sensitivity (upwards) and specificity (downwards) of regularized estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

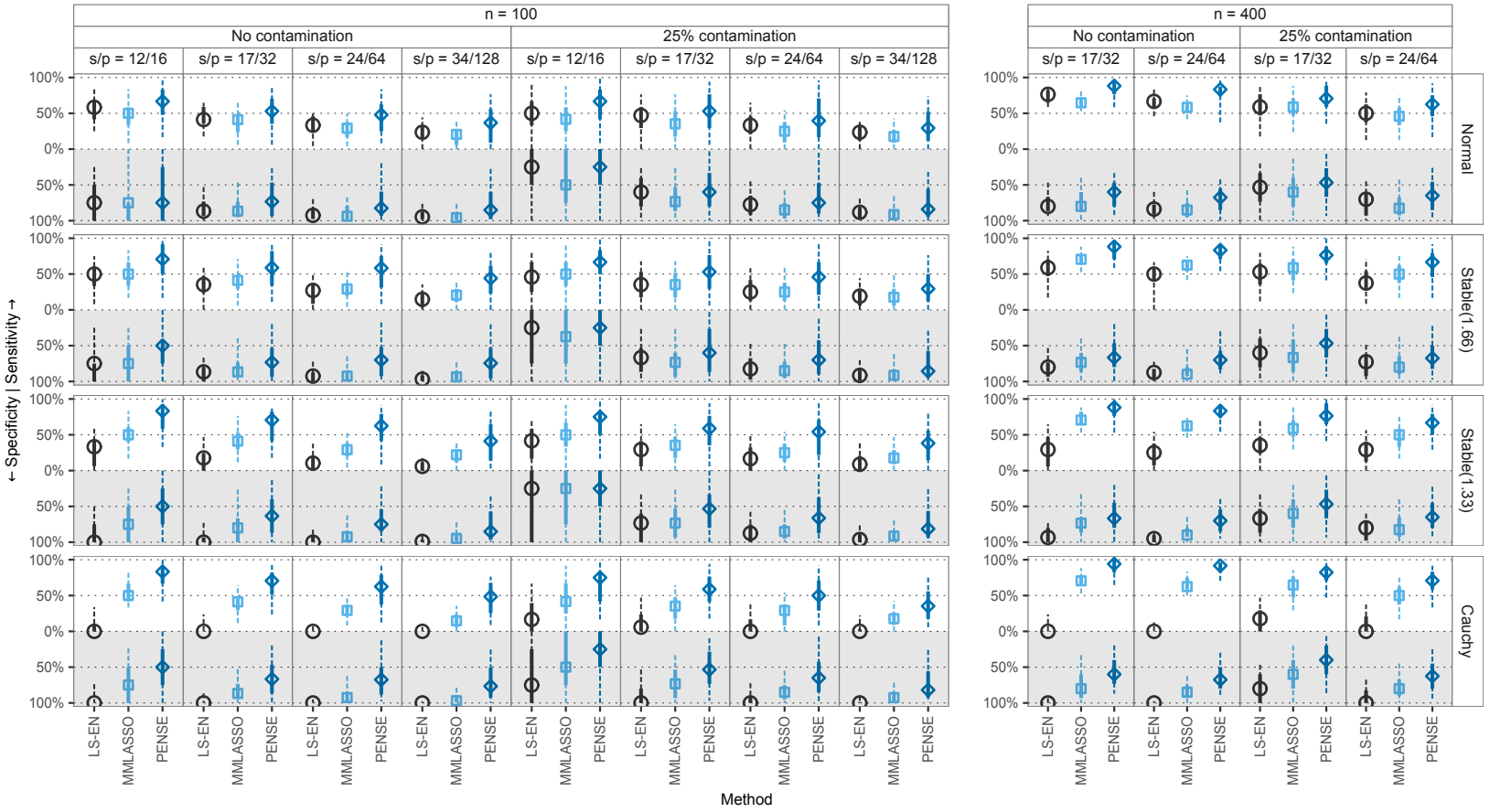


Figure C.4: Sensitivity (upwards) and specificity (downwards) of regularized estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

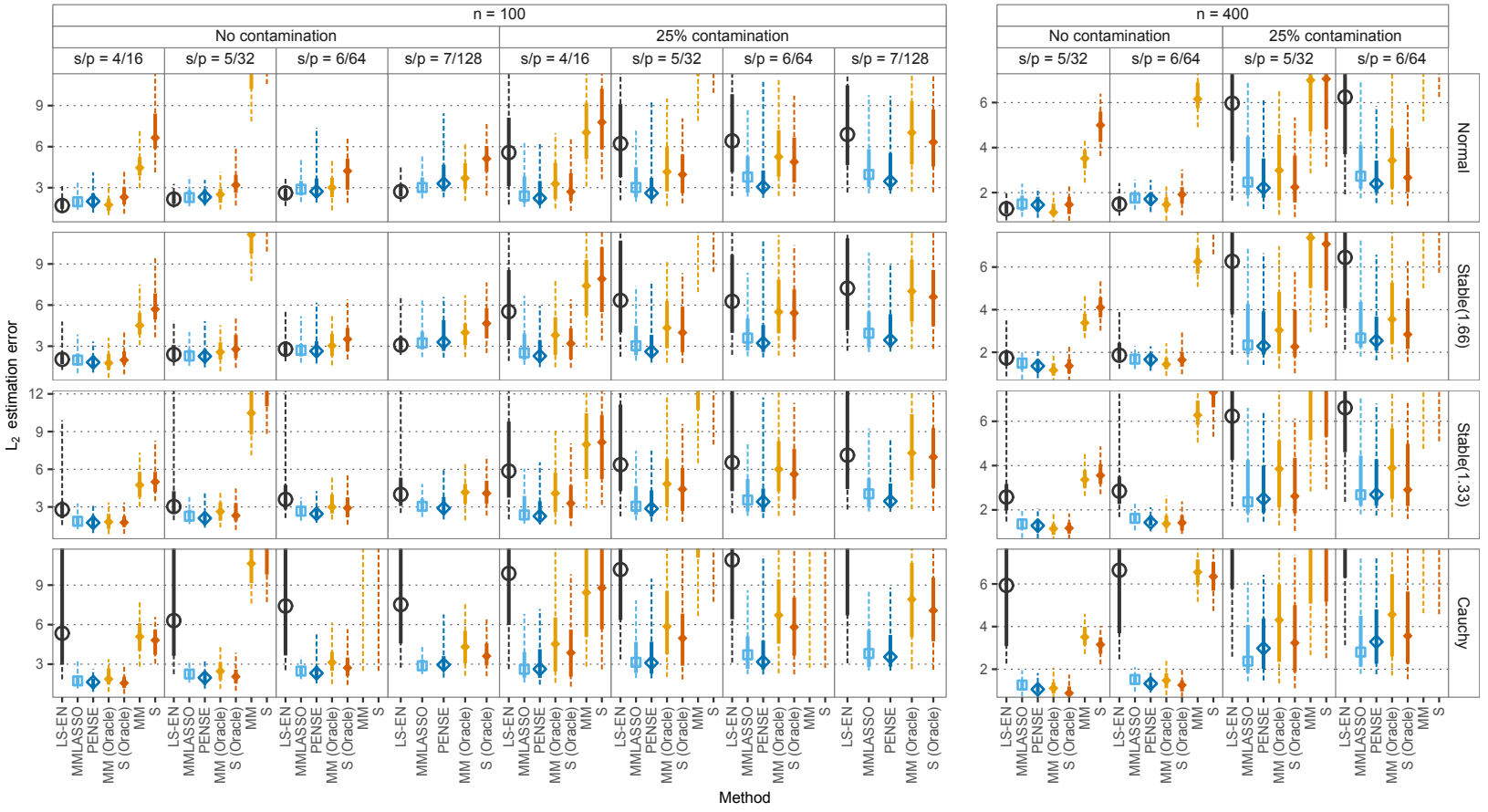


Figure C.5: Estimation accuracy in terms of the L_2 estimation error of several estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

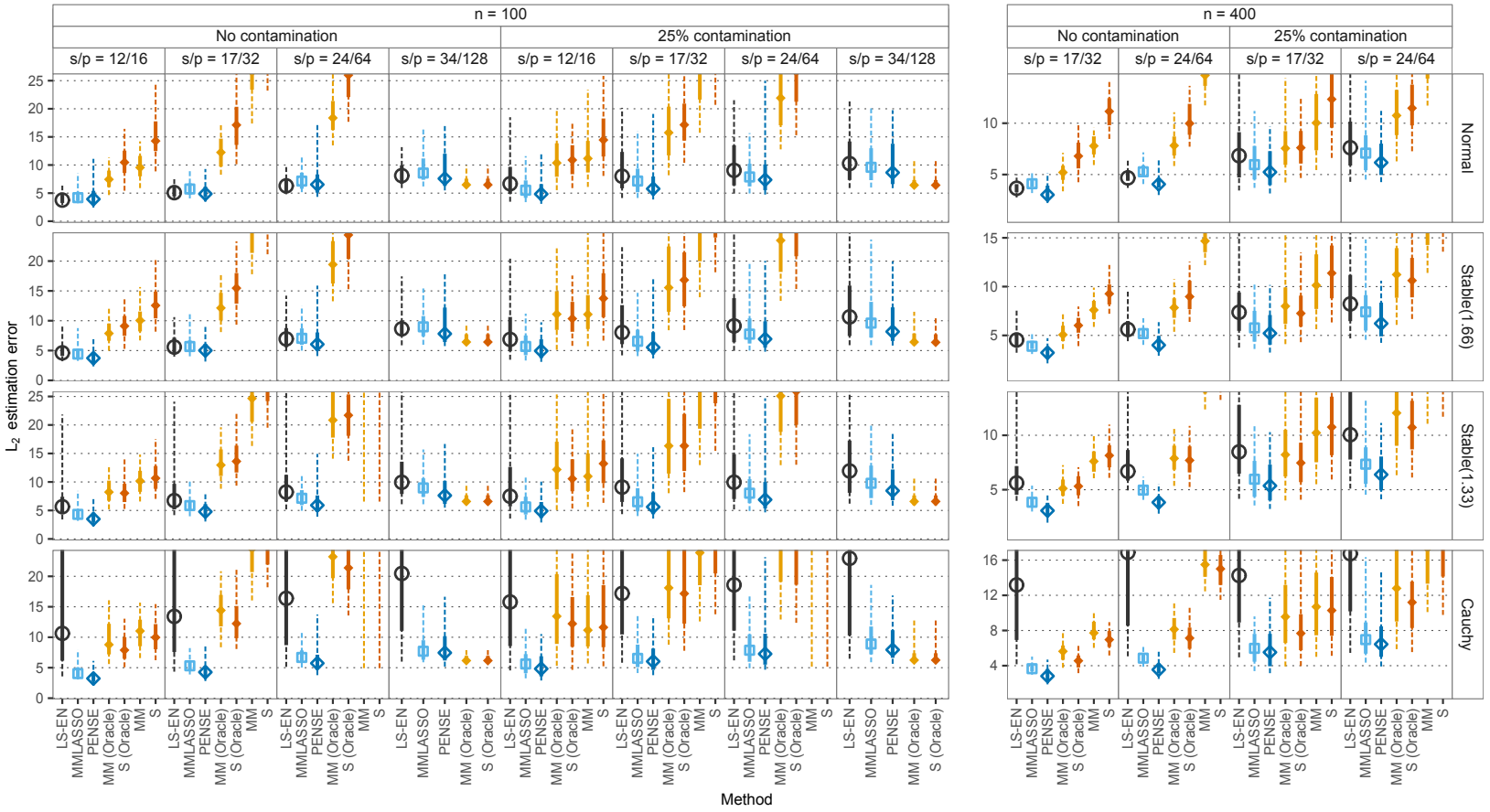


Figure C.6: Estimation accuracy in terms of the L_2 estimation error of several estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

C.2 Adaptive Elastic Net S-Estimators

Extending the numerical experiments from Section 3.6, below are detailed results for estimators discussed in Section 4.4 with the addition of other variable selection consistent estimators.

C.2.1 Prediction Performance

Prediction performance is measured in terms of the relative scale of the prediction error, as detailed in Section 3.6.3. Figures C.8 and C.9 show results for very sparse scenarios ($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

C.2.2 Variable Selection Performance

Variable selection performance is summarized by the sensitivity (i.e., the proportion of truly active predictors detected as such) and specificity (i.e., the proportion of truly inactive predictors detected as such). The summary figures show sensitivity and specificity in a single plot for regularized estimators only. Sensitivity extends upwards, specificity extends downwards. Methods perform well in terms of variable selection if the two points are at the top and bottom ends of the plot. Figures C.10 and C.11 show results for very sparse scenarios ($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

C.2.3 Estimation Accuracy

Estimation accuracy is assessed by the L_2 estimation error,

$$\text{RMSE}(\hat{\boldsymbol{\beta}}) = \sqrt{\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\right\|_2^2 + (\hat{\mu} - \mu^0)^2}.$$

The smaller the L_2 estimation error, the more accurate the estimation. Figures C.12 and C.13 show results for very sparse scenarios ($s = \log_2(p)$) and sparse scenarios ($s = 3\sqrt{p}$).

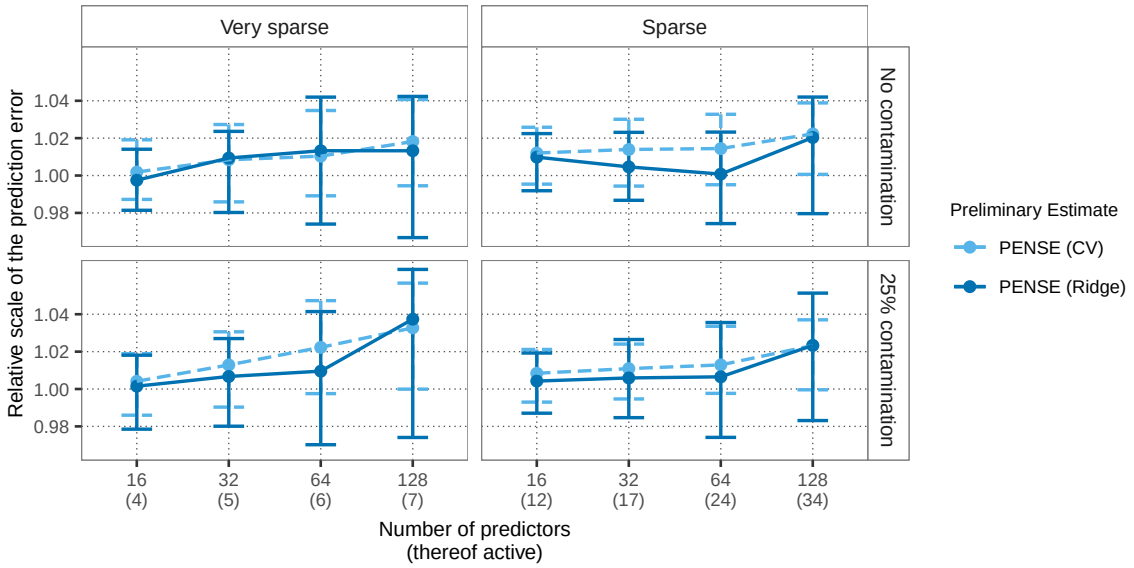


Figure C.7: Scale of the prediction error of adaptive PENSE, relative to the scale of the prediction error from PENSE. The preliminary estimates considered here are described in Section 4.4.1. Data is generated according to schemes *VS1-* (on the left) and *SP1-** (on the right) with $n = 100$ observations and 25% variance explained by the true model. In scenarios without contamination (top), plots show summaries of the metric over 400 values from 100 replications and 4 different error distributions. In scenarios introducing 25% contamination (bottom), plots show summaries of 1000 values from 50 replications of 5 different outlier positions and 4 different error distributions. The dots mark the median value, while error bars span the range of the inner 50%.

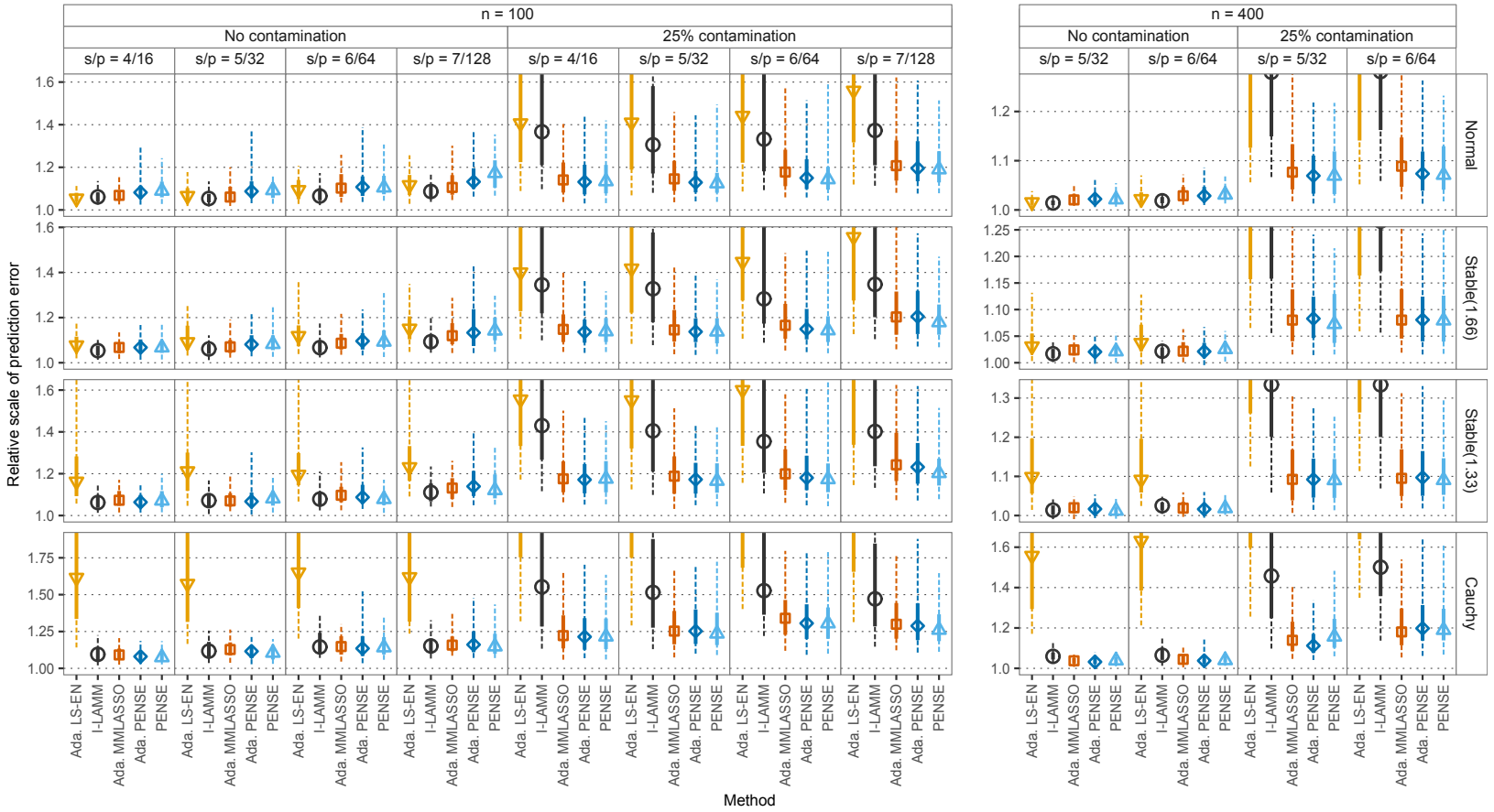


Figure C.8: Prediction performance of estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

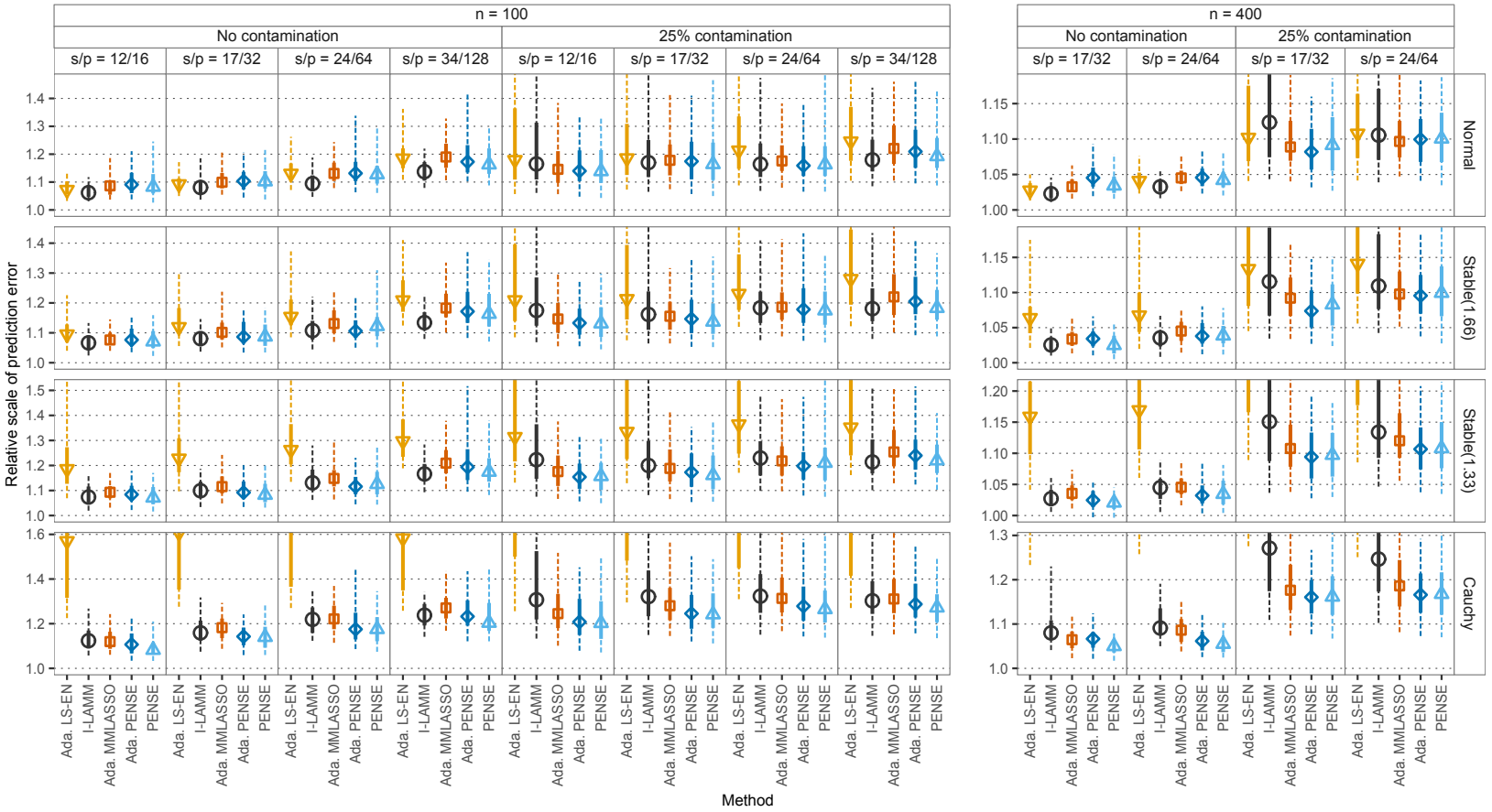


Figure C.9: Prediction performance of estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

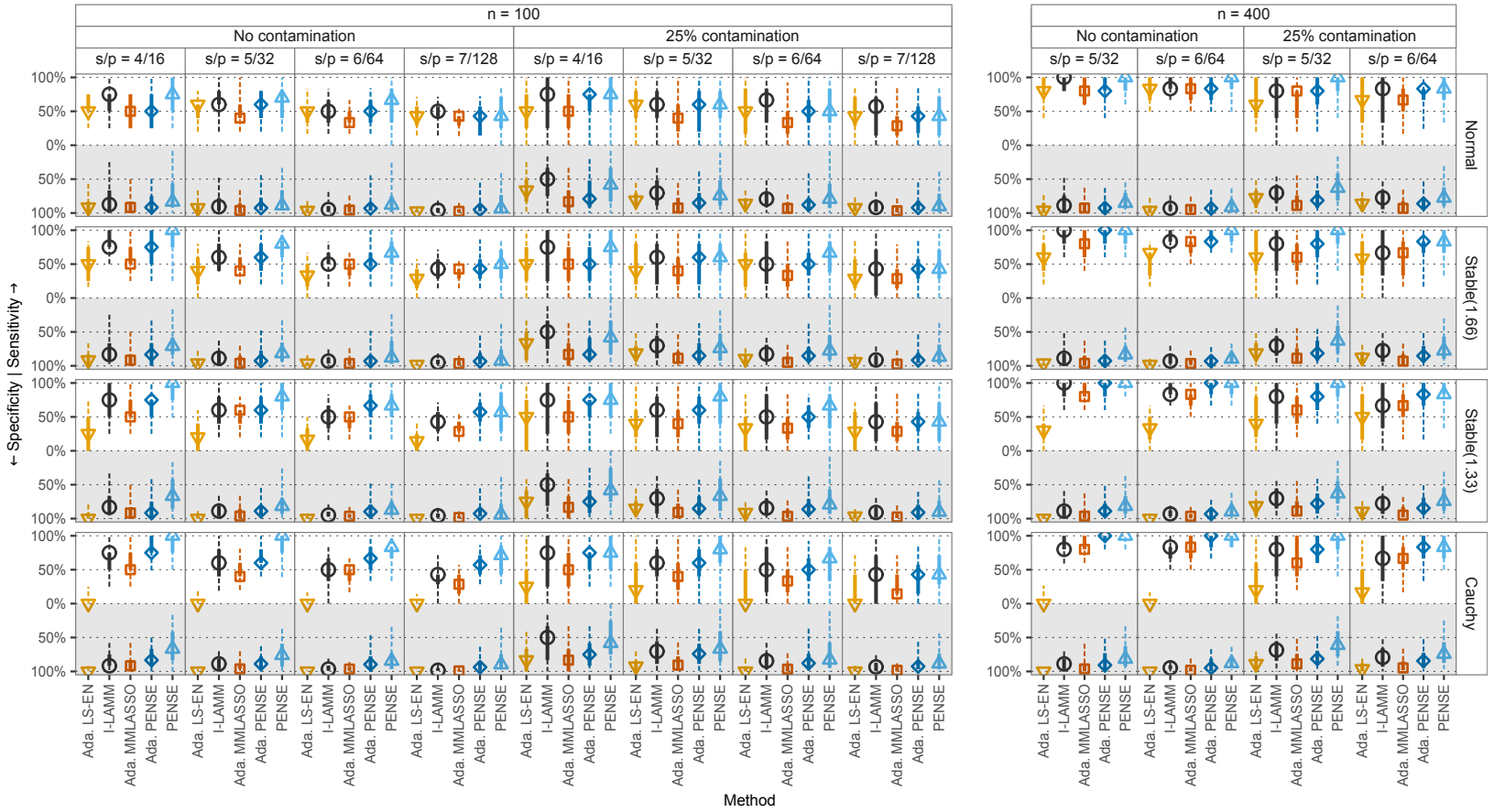


Figure C.10: Sensitivity (upwards) and specificity (downwards) of regularized estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

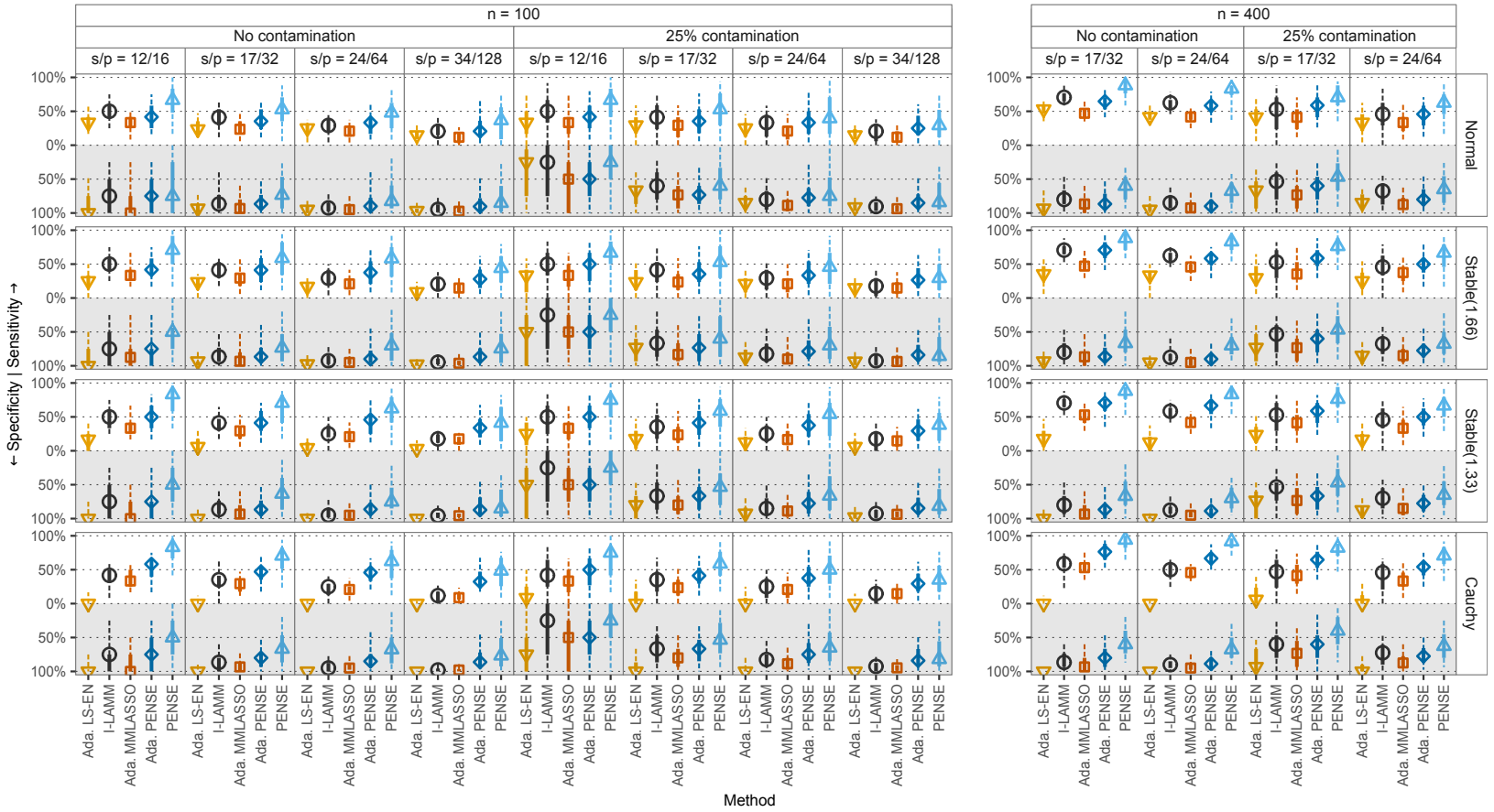


Figure C.11: Sensitivity (upwards) and specificity (downwards) of regularized estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

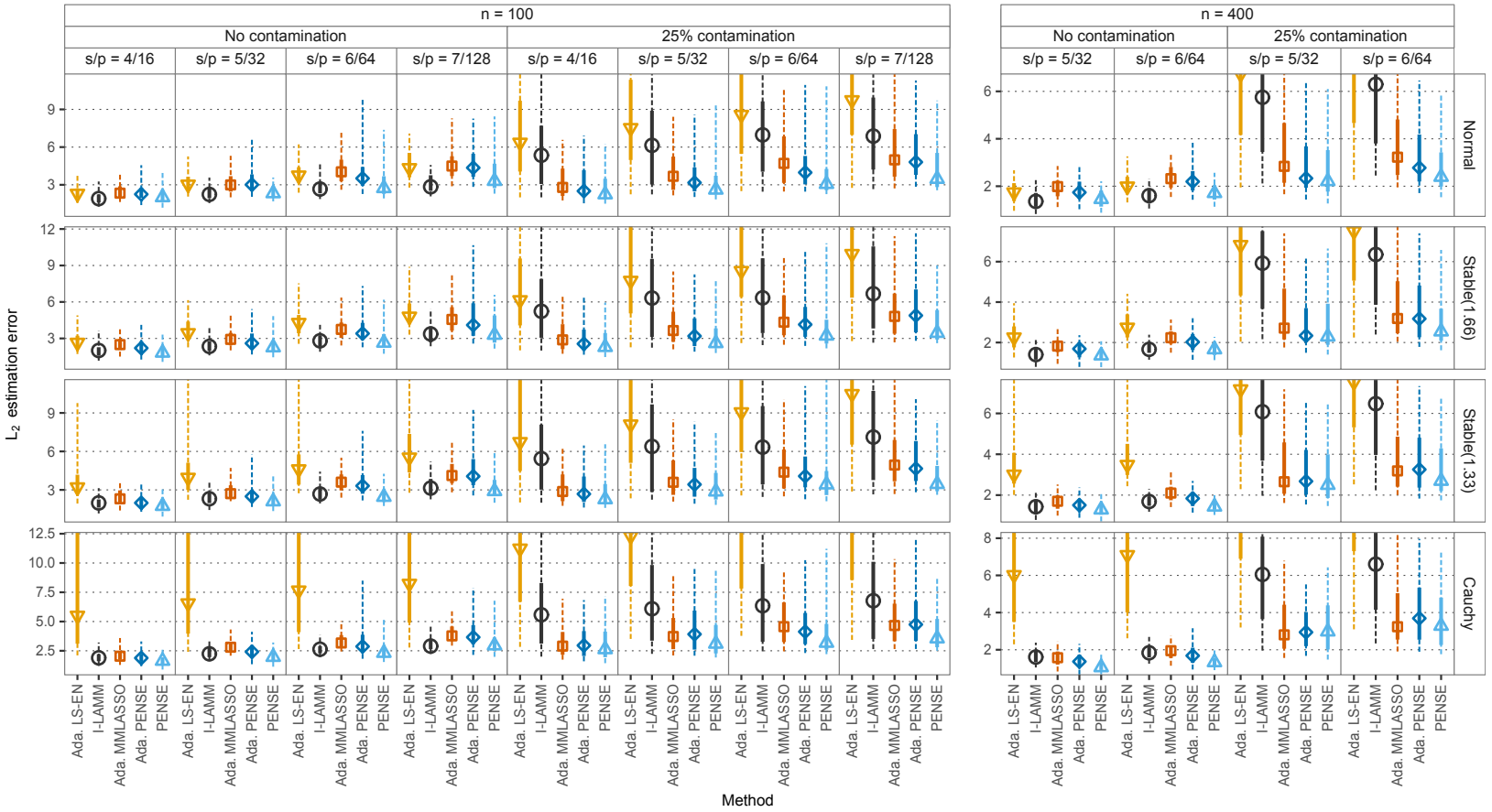


Figure C.12: Estimation accuracy in terms of the L_2 estimation error of several estimates under data generation scheme *VS1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.

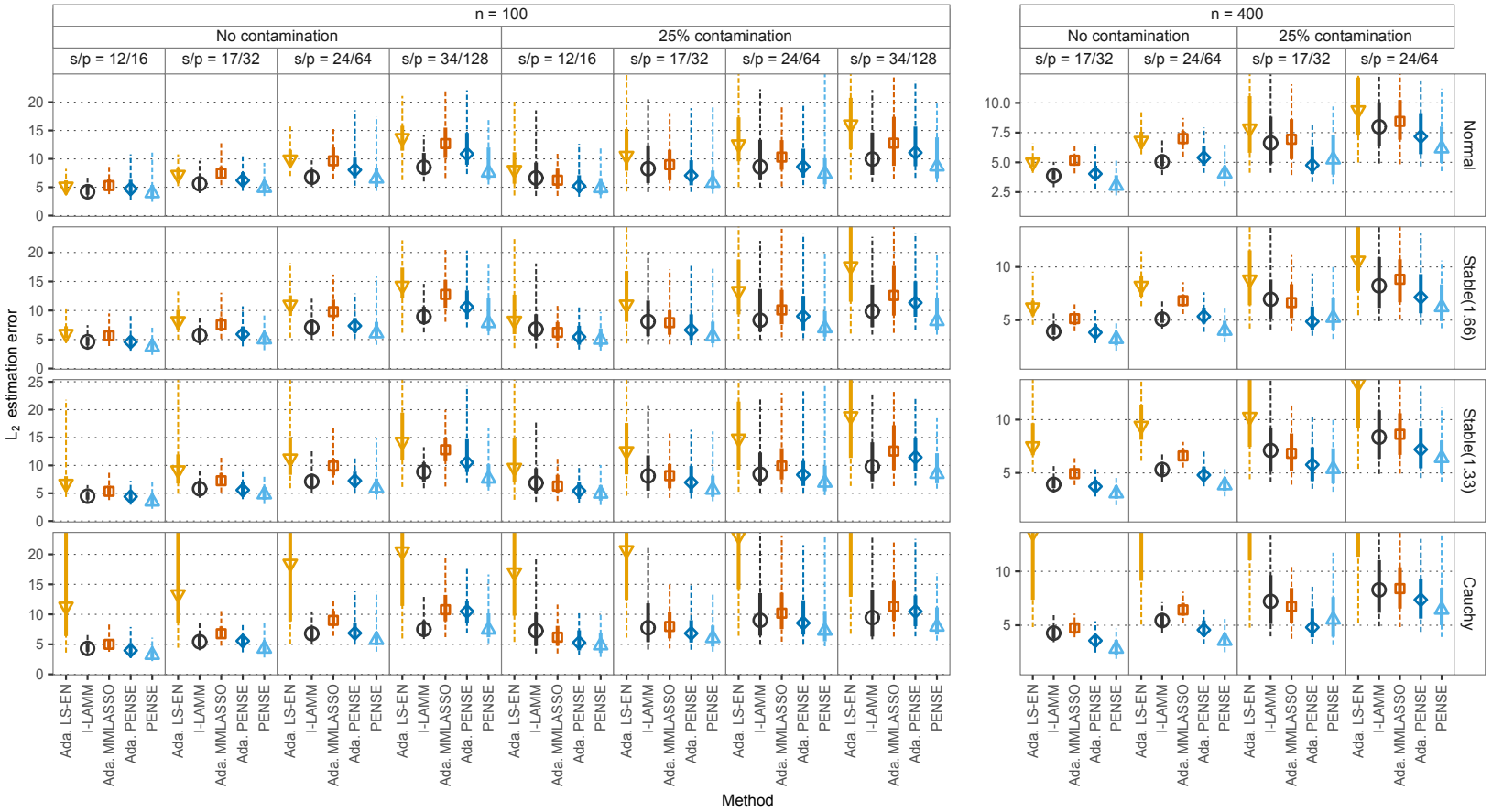


Figure C.13: Estimation accuracy in terms of the L_2 estimation error of several estimates under data generation scheme *SP1-**. In scenarios without contamination (left), plots show summaries of the metric over 100 replications. In scenarios introducing 25% contamination (right), plots show summaries of 250 values from 50 replications of 5 different outlier positions. The dots show the median value, while solid lines show the range of the inner 50% and the dashed whiskers extend from the 5% to the 95% quantile.