Fall 2016

# Penalized methods and algorithms for high-dimensional regression in the presence of heterogeneity

Congrui Yi
*University of Iowa*

# PENALIZED METHODS AND ALGORITHMS FOR HIGH-DIMENSIONAL REGRESSION IN THE PRESENCE OF HETEROGENEITY

by

Congrui Yi

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Statistics in the
Graduate College of The
University of Iowa

December 2016

Thesis Supervisor: Professor Jian Huang

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

---

PH.D. THESIS

---

This is to certify that the Ph.D. thesis of

Congrui Yi

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Statistics at the December 2016 graduation.

Thesis committee: _____
Jian Huang, Thesis Supervisor

_____
Patrick Breheny

_____
Kung-Sik Chan

_____
Joseph B. Lang

_____
Luke Tierney

## ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jian Huang, for his guidance and encouragement throughout my years of doctoral studies. He has introduced me to the fascinating field of high-dimensional statistics, and provided me enormous help in researches. To me, he has been a wonderful role model, an inspiring mentor, and a supportive friend. I would also like to thank the other members of my committee, Dr. Patrick Breheny, Dr. Kung-Sik Chan, Dr. Joseph B. Lang and Dr. Luke Tierney, for offering me valuable comments and advice within the dissertation process and during my graduate program. Finally, I also want to thank my Mom and Dad for always being there for me and encouraging me to pursue my interests, and my fiancée, Huan, for making me happy and supporting me through stressful moments.

## ABSTRACT

In fields such as statistics, economics and biology, heterogeneity is an important topic concerning validity of data inference and discovery of hidden patterns. This thesis focuses on penalized methods for regression analysis with the presence of heterogeneity in a potentially high-dimensional setting. Two possible strategies to deal with heterogeneity are: robust regression methods that provide heterogeneity-resistant coefficient estimation, and direct detection of heterogeneity while estimating coefficients accurately in the meantime.

We consider the first strategy for two robust regression methods, Huber loss regression and quantile regression with Lasso or Elastic-Net penalties, which have been studied theoretically but lack efficient algorithms. We propose a new algorithm Semismooth Newton Coordinate Descent to solve them. The algorithm is a novel combination of Semismooth Newton Algorithm and Coordinate Descent that applies to penalized optimization problems with both nonsmooth loss and nonsmooth penalty. We prove its convergence properties, and show its computational efficiency through numerical studies.

We also propose a nonconvex penalized regression method, Heterogeneity Discovery Regression (HDR) , as a realization of the second idea. We establish theoretical results that guarantees statistical precision for any local optimum of the objective function with high probability. We also compare the numerical performances of HDR with competitors including Huber loss regression, quantile regression and least squares through simulation studies and a real data example. In these experiments, HDR methods are able to detect heterogeneity accurately, and also largely outperform the competitors in terms of coefficient estimation and variable selection.

## PUBLIC ABSTRACT

In fields such as statistics, economics and biology, heterogeneity is an important topic concerning validity of data inference and discovery of hidden patterns. Our insights and interpretation of the data can be dramatically influenced by the presence of heterogeneity. And this is especially challenging in high-dimensional data which become increasingly common nowadays in many areas such as genetics, behavioral sciences, image and natural language processing. This thesis focuses on penalized methods for regression analysis with the presence of heterogeneity in a potentially high-dimensional setting.

One strategy to deal with heterogeneity is robust regression methods that provide heterogeneity-resistant coefficient estimation. We develop a novel algorithm, Semismooth Newton Coordinate Descent, that computes two important classes of penalized robust regression methods efficiently and scales very well to ultra-high dimensions (e.g. 100000). Another strategy is direct detection of heterogeneity while estimating coefficients accurately in the meantime. We propose a nonconvex penalized regression method, Heterogeneity Discovery Regression (HDR), as a realization of this idea. We establish good theoretical properties for the approach,and demonstrate significant advantages of HDR over alternatives such as robust regressions through simulation studies. Finally, we also illustrate the application of HDR to a building energy data.

## TABLE OF CONTENTS

## LIST OF TABLES

Table

## LIST OF FIGURES

Figure

# CHAPTER 1
# INTRODUCTION

In statistical data analysis, we almost always assume that each part of a dataset or population has the same statistical properties as any other part. But the validity of this homogeneity assumption is usually questionable, and heterogeneity is likely to arise in many scenarios. For biomedical studies, the same treatment or medicine can have different effects on different subjects. For spatial-temporal data, patterns and trends can vary geographically and change over time. And for integrative analysis (e.g. of genomic datasets), combining data from different sources is also subject to new problems such as heteroskedastic measure errors.

In this work, we are most interested in dealing with heterogeneity issues within the context of regression, where heterogeneity is usually driven from latent factors or unknown interactions between different factors. When we consider regression analysis in the high-dimensional setting, the problem becomes even more challenging because the heterogeneity problem is further complicated by the need for variable selection in order to achieve an interpretable model.

## 1.1 Thesis structure

Chapter 2 introduces mathematical foundations behind the penalized regression methods and optimization algorithms proposed in this thesis.

Compared to the commonly used least squares method, Robust regression methods such as Huber loss regression (Huber, 1973) and quantile regression (Koenker and Bassett Jr, 1978) provide more reliable coefficient estimation in the presence of heterogeneity. Penalized versions of these two methods have been discussed mainly in the theoretical aspect but efficient algorithms are not available. In Chapter 3, we consider the Lasso (Tibshirani, 1996) or Elastic-Net (Zou and Hastie, 2005) penalized versions of these methods and develop a new algorithm, Semismooth Newton

Coordinate Descent (SNCD), to solve them. We demonstrate the computational efficiency and scalability of SNCD through numerical experiments.

In Chapter 4, we adopt a different strategy to deal with heterogeneity. We propose a new class of methods, Heterogenity Discovery Regression (HDR), to provide not only trustworthy estimation similar to what robust regression methods are intended for, but also simultaneous detection of heterogeneity. We establish a finite-sample error bound that guarantees statistical precision for any local optimizer of the HDR objective. We also conduct an extensive set of numerical experiments to compare empirical performances of HDR methods with penalized robust regression and least squares. With the presence of heterogeneity, HDR methods have superior performances in both coefficient estimation and variable selection while they also automatically detect heterogeneity with high accuracy.

Finally, we summarize the thesis and discuss future work in Chapter 5.

# CHAPTER 2
# PRELIMINARIES

In this chapter we prepare several mathematical concepts and tools based on which this thesis is built. In particular, both Chapter 3 and 4 involve optimization with nonsmooth regularizers included in the objective functions, for which the concepts of subgradients are critical for building the optimality conditions. Besides, the Semismooth Newton Algorithm (SNA) is also closely related to the SNCD algorithm proposed in Chapter 3.

## 2.1 Subgradient and optimality

### 2.1.1 Convex function

For a convex function $f : \mathbb{R}^p \mapsto \mathbb{R}$, a vector $w \in \mathbb{R}^p$ is called a *subgradient* (Rockafellar, 1970) of $f$ at a point $z \in \mathbb{R}^p$ if

$$f(x) - f(z) \geq w^\top (x - z), \quad \forall x \in \mathbb{R}^p. \tag{2.1}$$

The set of all subgradients of $f$ at $z$ is called the *subdifferential*, denoted as $\partial f(z)$. When $f$ is differentiable at $z$, the subdifferential $\partial f(z) = \{\nabla f(z)\}$; but when $f$ is nondifferentiable at $z$, it is a closed set containing more than one element. For example, the subdifferential of the absolute value function has the following form

$$\partial |z| = \begin{cases} \{\text{sign}(z)\} & \text{if } z \neq 0, \\ [-1, 1] & \text{if } z = 0. \end{cases} \tag{2.2}$$

For convex optimization problems, the necessary and sufficient optimality conditions are called the KKT conditions. In the case of unconstrained optimization, the KKT conditions can be stated in terms of Fermat's rule (Rockafellar, 1970): for a convex function $f$,

$$\mathbf{0} \in \partial f(z^*) \Leftrightarrow z^* \in \arg\min_z f(z). \tag{2.3}$$

This holds because by definition $\mathbf{0} \in \partial f(z^*)$ if and only if for any $z$ we have $f(z) - f(z^*) \geq \mathbf{0}^\top (z - z^*) = 0$, i.e. $z^* \in \arg\min_z f(z)$.

A more general result (Combettes and Wajs, 2005) is

$$w \in \partial f(z) \Leftrightarrow z \in \text{Prox}_f(z + w), \tag{2.4}$$

where $\text{Prox}_f$ is the *proximity operator* for $f$ defined as

$$\text{Prox}_f(z) \coloneqq \arg\min_x \frac{1}{2}\|x - z\|_2^2 + f(x).$$

The second statement can be shown as follows. Applying Fermat's rule,

$$z \in \text{Prox}_f(z + w) = \arg\min_x \frac{1}{2}\|x - z - w\|_2^2 + f(x),$$

if and only if there exists $s \in \partial f(x)$ such that

$$\mathbf{0} = (z - z - w) + s = -w + s,$$

that is,

$$w = s \in \partial f(x).$$

Again, take the absolute value function $|\cdot|$ for example. It can be shown that its proximity operator is given in closed form by the soft-thresholding operator with threshold 1, i.e.

$$S(z) = \text{sgn}(z)(|z| - 1)_+. \tag{2.5}$$

### 2.1.2 General

For this part, we will introduce the conceptual extension to general nonsmooth functions.

Let $X$ and $Y$ be two metric spaces, then a function $f : X \mapsto Y$ is said to be *locally Lipschitz continuous* (Chen et al., 2000) at $z \in X$ if there exists $\delta > 0$,

$K > 0$ such that for any $x \in X$ that satisfies $\|x - z\| < \delta$, we have

$$\|f(x) - f(z)\| \leq K\|x - z\|.$$

If $f$ is convex or differentiable, then it is also locally Lipschitz continuous.

Now consider the case $X = \mathbb{R}^p, Y = \mathbb{R}$. Suppose $f$ is locally Lipschitz continuous at $z$, then for any direction $d$, we define the *directional derivative* of $f$ at $z$ in the direction $d$ by

$$f'(z; d) = \limsup_{t \downarrow 0} \frac{f(z + td) - f(z)}{t}, \tag{2.6}$$

which is finite due to the locally Lipschitz continuity.

Then a vector $w$ is called a (generalized) subgradient (Clarke, 1990) of function $f$ at $z$ if

$$f'(z; d) \geq w^\top d, \quad \forall d. \tag{2.7}$$

When $f$ is convex, it is easy to see this definition is equivalent with the previous one (2.1). And similarly, we call the set of all subgradients of $f$ at $z$ as the subdifferential and denote it by $\partial f(z)$. Linearity and chain rules for constructing subgradients follow easily from the definition.

We first consider the unconstrained optimization problem, $\min_z f(z)$. Note that $z^*$ is a local minimum of $f$ if and only if

$$f'(z^*; d) \geq 0, \quad \forall d,$$

which further implies

$$0 \in \partial f(z^*).$$

Interestingly, this condition is a necessary and sufficient condition when $f$ is convex but only a necessary condition in this general case.

Next consider the constrained optimization problem

$$\begin{aligned} \min_z \quad & f(z) \\ \text{subject to} \quad & g_i(z) \leq 0, \quad i = 1, \ldots, n \\ & h_j(z) = 0, \quad j = 1, \ldots, m \end{aligned} \tag{2.8}$$

where $f$, $g_i$, $h_j$ are locally Lipschitz continuous near any point in $\mathbb{R}^p$. Denote $g = (g_1, \ldots, g_n)^\top$, $h = (h_1, \ldots, h_m)$. Then the Lagrangian $L(z, \lambda, r, s) : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ is defined by

$$L(z, \lambda, r, s) = \lambda f(z) + r^\top g(z) + s^\top h(z).$$

Using $u \succeq v$ to denote that $u_i \geq v_i, i = 1, \ldots, p$ for $u, v \in \mathbb{R}^p$, we have the following result

**Theorem 2.1.** *(Necessary Conditions) Let $z^*$ be a local minimizer of* (2.8)*. Then there exist $\lambda^* \geq 0$, $r^* \succeq \mathbf{0}$, and $s^*$, not all zero, such that $r^{*\top} g(z^*) = 0$ and $\mathbf{0} \in \partial_z L(z^*, \lambda^*, r^*, s^*)$.*

The necessary conditions in Theorem 2.1 can be viewed as degenerate when $\lambda^* = 0$, since the function $f$ being minimized is not involved. Various "constraint qualification" conditions have been proposed to ensure that the theorem holds in the "normal" form with $\lambda^* = 1$. For our purpose, we will introduce a particular example, the so-called Slater conditions. For a discussion of constraint qualifications in more details, see Chapter 6 of Clarke (1990).

Suppose the optimization problem (2.8) has no equality constraints, i.e. $m = 0$, and that $g_i$'s are convex. In this context, the *Slater condition* is the assumption that there exists a strictly feasible point $\widehat{z} \in \mathbb{R}^p$, i.e. $g_i(\widehat{z}) < 0, i = 1, \ldots, n$.

Next, we define $M_\lambda(z)$, the index $\lambda$ multiplier set corresponding to a feasible point $z$, to be the set of vectors $r$ (no $s$ because we assume $m = 0$) satisfying the

conclusions of Theorem 2.1 together with $\lambda$. That is,

$$M_\lambda(z) = \left\{ r \in \mathbb{R}^n : \mathbf{0} \in \partial_z L(z, \lambda, r), r^\top g(z) = 0, r \succeq \mathbf{0} \right\},$$

where $\lambda \geq 0$. Then we have

**Theorem 2.2.** *If the Slater condition holds in the problem described above, then* $M_0(z)$ *reduces to* $\{\mathbf{0}\}$ *for any feasible* $z$*.*

This result immediately implies that any $\lambda$ value satisfying the conclusions of Theorem 2.1 must be positive, hence we can rescale $\lambda$ and $r$ so that $\lambda = 1$.

## 2.2 Newton derivatives and semismooth Newton algorithm

Based on the concepts of generalized Jacobian (Clarke, 1983) and semismoothness (Mifflin, 1977), Qi and Sun (1993) established superlinear convergence of a Newton-type method for solving finite-dimensional nonsmooth equations, hence the name Semismooth Newton Algorithm (SNA). The Newton differentiability was introduced later for more general problems including infinite-dimensional cases (Chen et al., 2000; Ito and Kunisch, 2008). It has a simpler formulation and is actually a milder condition than semismoothness. Newton derivatives can be calculated via basic algebra and chain rules as indicated in Lemmas A.1, A.2 and A.3 in Section A in the Supplementary Materials.

**Definition 2.1.** A function $F : \mathbb{R}^m \to \mathbb{R}^l$ is said to be *Newton differentiable* at $z \in \mathbb{R}^m$ if there exists an open neighborhood $\mathcal{N}(z)$ and a mapping $H : \mathcal{N}(z) \to \mathbb{R}^{l \times m}$ such that $\{H(z+h) : z + h \in \mathcal{N}(z), h \neq 0\}$ is uniformly bounded in spectral norm induced by the Euclidean norm and

$$\|F(z+h) - F(z) - H(z+h)h\|_2 = o(\|h\|_2) \quad \text{as } h \to 0.$$

Here $H$ is called a *Newton derivative* for $F$ at $z$. The set of all Newton derivatives at $z$ is denoted as $\nabla_N F(z)$.

**Remark.** (1) Unlike other notions of differentiability, "$H(z)$" does not appear in the above definition, so in general, for a Newton derivative $H$ for $F$ at $z$, $H(z)$ is not characterized by a limit of quotients. Actually, it can take any value.

(2) The requirement that $H$ is uniformly bounded in the neighborhood of $z$ is very important. Otherwise, any function $F$ is Newton differentiable at $z$ with $H(z+h) = (F(z+h) - F(z))h^\top / \|h\|_2^2$ as a Newton derivative.

Recall that in the last section we introduce the concept of locally Lipschitz continuity. It turns out $F$ is Newton differentiable at $z$ if and only if $F$ is locally Lipschitz continuous at $z$ (Chen et al., 2000). This gives a simple characterization of the Newton differentiability and shows that Newton differentiability defines a much wider class of functions than the usual concept of differentiability.

Now we provide several properties useful for calculating Newton derivatives. The first one is the following chain rule for Newton derivatives (Ito and Kunisch, 2008).

**Lemma 2.3.** *If $F : \mathbb{R}^l \to \mathbb{R}^m$ is continuously Fréchet differentiable at $z \in \mathbb{R}^l$ with Jacobian $J_F$ and $G : \mathbb{R}^m \to \mathbb{R}^n$ is Newton differentiable at $F(z)$ with a Newton derivative $H_G$. Then $G \circ F$ is Newton differentiable at $z$ with a Newton derivative $H_G(F(z+h))J_F(z+h)$ for $h$ sufficiently small.*

We also derived two other results.

**Lemma 2.4.** *In the following, assume $F : \mathbb{R}^m \to \mathbb{R}^l$, $G : \mathbb{R}^m \to \mathbb{R}^l$, $z \in \mathbb{R}^m$, $F = (F_1, \ldots, F_l)^\top$ and $H = (H_1^\top, \ldots, H_l^\top)^\top$, where $H_i \in \mathbb{R}^{1\times m}, \quad i = 1, \ldots, l$.*

*(i) If $F$ is continuously Fréchet differentiable at $z$, then $F$ is also Newton differentiable at $z$ and $J_F \in \nabla_N F(z)$;*

*(ii) If $F$ is Newton differentiable at $z$, then for any integer $k > 0$ and $A \in \mathbb{R}^{k\times l}$, $AF$ is Newton differentiable at $z$; if $H \in \nabla_N F(z)$, then $AH \in \nabla_N AF(z)$;*

*(iii) If $F$ and $G$ are Newton differentiable at $z$, then $F+G$ is Newton differentiable*

*at $z$; if $H_F \in \nabla_N F(z)$, $H_G \in \nabla_N G(z)$, then $H_F + H_G \in \nabla_N (F+G)(z)$;*

*(iv) $F$ is Newton differentiable at $z$ if and only if $F_1, \ldots, F_l$ are all Newton differentiable at $z$ and $H \in \nabla_N F(z) \Leftrightarrow H_i \in \nabla_N F_i(z), \quad i = 1, \ldots, l$;*

**Lemma 2.5.** *A univariate piecewise-smooth real function $f$ is everywhere Newton differentiable, with a Newton derivative $H$ given by*

$$H(z) = \begin{cases} f'(z) & \text{if } f \text{ is differentiable at } z, \\ \text{arbitrary real number} & \text{if } f \text{ is not differentiable at } z. \end{cases}$$

The following result due to Chen et al. (2000) establishes the superlinear convergence of SNA under the Newton differentiability.

**Theorem 2.6.** *Suppose that $F : \mathbb{R}^m \to \mathbb{R}^m$ is Newton differentiable at a solution $z^*$ of $F(z) = \mathbf{0}$. Let $H$ be a Newton derivative for $F$ at $z^*$. Suppose there exists a neighborhood $\mathcal{N}(z^*)$ and $M > 0$ such that $H(z)$ is nonsingular and $\|H(z)^{-1}\| \leq M$ for all $z \in \mathcal{N}(z^*)$, then the Newton-type iteration*

$$z^{k+1} = z^k - H(z^k)^{-1} F(z^k), \ k = 0, 1, \ldots$$

*converges superlinearly to $z^*$ provided that $\|z^0 - z^*\|_2$ is sufficiently small, where $z^0$ is the initial value.*

# CHAPTER 3
# SEMISMOOTH NEWTON COORDINATE DESCENT (SNCD) FOR TWO PENALIZED ROBUST REGRESSION METHODS

## 3.1 Introduction

Consider the linear regression model

$$y_i = \beta_0 + x_i^\top \beta + \varepsilon_i$$

where $x_i$ is a $p$-dimensional vector of covariates, $(\beta_0, \beta)$ are regression coefficients, and $\varepsilon_i$ is the random error independent of $x_i$. We are interested in the high dimensional case where $p \gg n$ and the model is sparse in the sense that only a small proportion of the coefficients are nonzero. In such a scenario, a key task is identifying and estimating the nonzero coefficients. A popular approach is the penalized regression

$$\min_{\beta_0,\beta} \frac{1}{n} \sum_i \ell(y_i - \beta_0 - x_i^\top \beta) + \lambda P(\beta), \tag{3.1}$$

where $\ell$ is a generic loss function and $p$ is a penalty function with a tuning parameter $\lambda \geq 0$. We consider the elastic-net penalty (Zou and Hastie, 2005)

$$P(\beta) \equiv P_\alpha(\beta) = \alpha\|\beta\|_1 + (1-\alpha)\frac{1}{2}\|\beta\|_2^2, 0 \leq \alpha \leq 1,$$

which is a convex combination of the lasso (Tibshirani, 1996) ($\alpha = 1$) and the ridge penalty (Hoerl and Kennard, 1970) ($\alpha = 0$).

A common choice for $\ell$ is the squared loss $\ell(t) = t^2/2$, corresponding to the least squares regression in classical regression literature. Although the squared loss is analytically simple, it is not suitable for data in the presence of outliers or heterogeneity. Instead, we could consider two widely used robust alternatives, the Huber loss (Huber, 1973) and the quantile loss (Koenker and Bassett Jr, 1978).

The Huber loss is

$$\ell(t) \equiv h_\gamma(t) = \begin{cases} \frac{t^2}{2\gamma}, & \text{if } |t| \le \gamma, \\ |t| - \frac{\gamma}{2}, & \text{if } |t| > \gamma, \end{cases} \tag{3.2}$$

where $\gamma > 0$ is a given constant. This function is quadratic for $|t| \le \gamma$ and linear for $|t| > \gamma$. In addition, it is convex and first-order differentiable. These features allow it to combine analytical tractability of the squared loss for the least squares and outlier-robustness of the absolute loss for the LAD regression.

The quantile loss is

$$\ell(t) \equiv \rho_\tau(t) = t(\tau - I(t < 0)), t \in \mathbb{R}, \tag{3.3}$$

where $0 < \tau < 1$. This is a generalization of the absolute loss with $\tau = 1/2$. Rather than the conditional mean of the response given the covariates, quantile regression models conditional quantiles. For heterogeneous data, the functional relationship between the response and the covariates may vary in different segments of its conditional distribution. By choosing different $\tau$, quantile regression provides a powerful technique for exploring data heterogeneity in addition to outlier-robustness.

Holland and Welsch (1977) proposed an iteratively re-weighted least squares algorithm for the unpenalized Huber loss regression. However, this algorithm does not have a natural extension to the penalized version. For unpenalized quantile regression, Portnoy et al. (1997) formulated its dual form as a linear programming problem and proposed an interior point method to solve it. The lasso penalized version can be shown to have a similar dual form, except that it becomes $(n + p)$-dimensional with $p$ extra constraints due to the penalty. Thus it can be solved using the same algorithm and this extension was implemented in the R package `quantreg` (`http://cloud.r-project.org/package=quantreg`). However, it is not clear if this approach is scalable to high-dimensional problems. Osborne and Turlach (2011) proposed a homotopy algorithm for computing solution paths of the constrained

version of the $L_1$ penalized quantile regression, which is not directly comparable with the unconstrained formulation considered here.

In recent years coordinate descent algorithms have proven to be very effective for pathwise optimization of penalized regression models, see for example, Friedman et al. (2007) for lasso and fused lasso penalized least squares, Friedman et al. (2010) for elastic-net penalized GLM, and Breheny and Huang (2011) for nonconvex penalized least squares and logistic regression. The loss functions considered by these authors are either quadratic, or twice differentiable which can be approximated quadratically via Taylor expansion. Hence the coordinate descent iterations have close-form solutions. However, the Huber loss is only first-order differentiable and the quantile loss is nondifferentiable, hence the above approach does not work. Wu and Lange (2008) proposed a greedy coordinate descent algorithm for lasso penalized LAD regression that amounts to computing a weighted median at each iteration, but the authors acknowledged that it could possibly converge to an inferior point as illustrated by a counterexample from Li and Arce (2004). Recently Peng and Wang (2015) proposed a QICD algorithm for nonconvex penalized quantile regression that iterates by first majorizing the penalty function and then solving the problem with coordinate descent. But when the lasso penalty is used, which does not need to be majorized, the algorithm becomes exactly the same as the one in Wu and Lange (2008). Besides, it seems neither algorithm can be easily generalized to the elastic-net penalty with $0 < \alpha < 1$.

In this chapter, we propose a novel semismooth Newton coordinate descent (SNCD) algorithm for computing solution paths of the elastic-net penalized Huber loss regression and quantile regression. This algorithm combines the coordinate descent algorithm with the semismooth Newton algorithm (SNA) for solving nonsmooth equations. It is highly efficient and scalable in high-dimensional settings. Unlike a typical coordinate descent method which only updates the primal variable

$\beta$, the SNCD utilizes both the primal and the dual information (via subgradient) in its iterations. In addition, an adaptive version of the strong rule (Tibshirani et al., 2012) for screening predictors is incorporated to gain extra efficiency. We also provide an implementation of SNCD through a publicly available R package `hqreg` (`http://cloud.r-project.org/package=hqreg`) which currently supports the Huber loss, the quantile loss and the squared loss. This algorithm can be generalized to other problems with nonsmooth loss functions, like the linear support vector machine with the hinge loss.

The rest of this chapter is organized as follows. In Section 3.2 we introduce SNCD for the penalized Huber loss regression and establish its convergence. In Section 3.3 we extend SNCD to penalized quantile regression. Section 3.4 describes the adaptive strong rule. Then at the end, we investigate the performance of `hqreg`, our implementation of SNCD, through simulation studies and real datasets.

## 3.2 SNCD for penalized Huber loss regression

Consider the Huber loss $\ell = h_\gamma$, then (3.1) becomes

$$\min_{\beta_0,\beta} f_H(\beta_0, \beta) = \frac{1}{n} \sum_i h_\gamma(y_i - \beta_0 - x_i^\top \beta) + \lambda P_\alpha(\beta). \tag{3.4}$$

Fix $\lambda$ and $\alpha$, and denote the optimizer by $(\widehat{\beta}_0, \widehat{\beta})$. Since the objective function in (3.4) is convex, $(\widehat{\beta}_0, \widehat{\beta})$ satisfies the necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions. Let $\partial|t|$ denote the set of subgradients of the absolute value function $|\cdot|$ at $t$, then using the concept of proximity operator discussed in Chapter 2, it can be shown that

$$s \in \partial|t| \text{ if and only if } t = S(t + s), \tag{3.5}$$

where $S$ is the soft-thresholding operator with threshold 1, i.e. $S(z) = \text{sgn}(z)(|z| - 1)_+$. As shown in Appendix A, combining this fact with some basic convex analysis

concepts introduced in Chapter 2, the KKT conditions of (3.4) can be written as

$$\begin{cases} -\frac{1}{n}\sum_i h'_\gamma(y_i - \widehat{\beta}_0 - x_i^\top \widehat{\beta}) = 0, \\ -\frac{1}{n}\sum_i h'_\gamma(y_i - \widehat{\beta}_0 - x_i^\top \widehat{\beta}) x_{ij} + \lambda\alpha\widehat{s}_j + \lambda(1-\alpha)\widehat{\beta}_j = 0, \\ \widehat{\beta}_j - S(\widehat{\beta}_j + \widehat{s}_j) = 0, \quad j = 1, \ldots, p, \end{cases} \tag{3.6}$$

where $\widehat{s}_j \in \partial|\widehat{\beta}_j|$ and $h'_\gamma(\cdot)$, the derivative of $h_\gamma(\cdot)$, is given by

$$h'_\gamma(t) = \begin{cases} \frac{t}{\gamma}, & \text{if } |t| \le \gamma, \\ \operatorname{sgn}(t), & \text{if } |t| > \gamma. \end{cases} \tag{3.7}$$

In this way the optimization problem (3.4) is transformed into a root finding problem for a system of nonsmooth equations (3.6). A straightforward approach is applying SNA to the entire system of equations. As discussed later in section 3.2.2, this approach contains many matrix operations that cause $O(np^2)$ computational cost per iteration, which severely limits its scalability.

For better efficiency and scalability, we propose a new algorithm, Semismooth Newton Coordinate Descent (SNCD), that combines SNA with cyclic coordinate descent in solving these equations. Similar to the Gauss-Seidel method for linear equations, SNCD solves the equations of (3.6) in a cyclic fashion to avoid cumbersome matrix operations. We cycle through $(\beta_0, \beta, s)$ in a pairwise fashion: at each step, a pair $(\beta_j, s_j)$ (and $\beta_0$ by itself) is updated by solving the corresponding part of (3.6), while the other variables are fixed at their current values $\tilde{\beta}_k, \tilde{s}_k, k \neq j$. Specifically, we solve the following equations at each step:

- For $(\beta_j, s_j)$:

$$\begin{cases} -\frac{1}{n}\sum_i h'_\gamma(\tilde{r}_i + x_{ij}\tilde{\beta}_j - x_{ij}\beta_j) x_{ij} + \lambda\alpha s_j + \lambda(1-\alpha)\beta_j = 0, \\ \beta_j - S(\beta_j + s_j) = 0, \end{cases} \tag{3.8}$$

- For $\beta_0$:

$$-\frac{1}{n}\sum_i h'_\gamma(\tilde{r}_i + \tilde{\beta}_0 - \beta_0) = 0, \tag{3.9}$$

where $\tilde{r}_i = y_i - \tilde{\beta}_0 - x_j^\top \tilde{\beta}, \quad i = 1, \ldots, n.$

Note that (3.8) is exactly the KKT conditions of

$$\min_{\beta_j} f_H(\ldots, \tilde{\beta}_{j-1}, \beta_j, \tilde{\beta}_{j+1}, \ldots),$$

and (3.9) the KKT condition of

$$\min_{\beta_0} f_H(\beta_0, \tilde{\beta}_1, \ldots).$$

Hence SNCD can be seen as a special type of coordinate descent.

Denote

$$\psi_\gamma(t) = \frac{1}{\gamma} I(|t| \leq \gamma), \tag{3.10}$$

then $\psi_\gamma \in \nabla_N h'_\gamma(t), \forall t \in \mathbb{R}$. The SNCD iterations proceed as follows:

(i) Updating $\beta_0$. Let

$$F_0(z; \tilde{\beta}) = -\frac{1}{n} \sum_i h'_\gamma(\tilde{r}_i + \tilde{\beta}_0 - z).$$

Since

$$H_0(z) = \frac{1}{n} \sum_i \psi_\gamma(\tilde{r}_i + \tilde{\beta}_0 - z) \in \nabla_N(F_0(z)),$$

we update $\beta_0$ by solving (3.9) via SNA

$$\beta_0 \leftarrow \tilde{\beta}_0 - H_0(\tilde{\beta}_0)^{-1} F_0(\tilde{\beta}_0) = \tilde{\beta}_0 + \frac{\sum_i h'(\tilde{r}_i)}{\sum_i \psi_\gamma(\tilde{r}_i)}.$$

(ii) Updating $(\beta_j, s_j)$. Let

$$F_j(z; \tilde{\beta}) = \begin{bmatrix} -\frac{1}{n} \sum_i h'_\gamma(\tilde{r}_i + x_{ij}\tilde{\beta}_j - x_{ij} z_1) x_{ij} + \lambda\alpha z_2 + \lambda(1-\alpha) z_1 \\ z_1 - S(z_1 + z_2) \end{bmatrix},$$

where $z = (z_1, z_2)^\top$. Since

$$z_1 - S(z_1 + z_2) = \begin{cases} -z_2 + \operatorname{sgn}(z_1 + z_2) & \text{if } |z_1 + z_2| > 1, \\ z_1 & \text{if } |z_1 + z_2| \leq 1, \end{cases} \tag{3.11}$$

we solve for $(\beta_j, s_j)$ from (3.8) via SNA in two types of updates:

(a) $|\tilde{\beta}_j + \tilde{s}_j| > 1$. For $z$ with $|z_1 + z_2| > 1$, a Newton derivative of $F_j$ at $z$ is

$$H_j(z) = \begin{bmatrix} \frac{1}{n}\sum_i \psi_\gamma(\tilde{r}_i + x_{ij}\tilde{\beta}_j - x_{ij}z_1)x_{ij}^2 + \lambda(1-\alpha) & \lambda\alpha \\ 0 & -1 \end{bmatrix} \in \nabla_N F_j(z). \tag{3.12}$$

Hence the update is

$$\begin{bmatrix} \beta_j \\ s_j \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{\beta}_j \\ \tilde{s}_j \end{bmatrix} - H_j(\tilde{\beta}_j, \tilde{s}_j)^{-1} F_j(\tilde{\beta}_j, \tilde{s}_j)$$
$$= \begin{bmatrix} \tilde{\beta}_j + \frac{\frac{1}{n}\sum_i h'_\gamma(\tilde{r}_i)x_{ij} - \lambda\alpha\,\mathrm{sgn}(\tilde{\beta}_j + \tilde{s}_j) - \lambda(1-\alpha)\tilde{\beta}_j}{\frac{1}{n}\sum_i \psi_\gamma(\tilde{r}_i)x_{ij}^2 + \lambda(1-\alpha)} \\ \mathrm{sgn}(\tilde{\beta}_j + \tilde{s}_j) \end{bmatrix}.$$

(b) $|\tilde{\beta}_j + \tilde{s}_j| \leq 1$. For $z$ with $|z_1 + z_2| \leq 1$, a Newton derivative of $F_j$ at $z$ is

$$H_j(z) = \begin{bmatrix} \frac{1}{n}\sum_i \psi_\gamma(\tilde{r}_i + x_{ij}\tilde{\beta}_j - x_{ij}z_1)x_{ij}^2 + \lambda(1-\alpha) & \lambda\alpha, \\ 1 & 0 \end{bmatrix} \in \nabla_N F_j(z). \tag{3.13}$$

Hence the update is

$$\begin{bmatrix} \beta_j \\ s_j \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{\beta}_j \\ \tilde{s}_j \end{bmatrix} - H_j(\tilde{\beta}_j, \tilde{s}_j)^{-1} F_j(\tilde{\beta}_j, \tilde{s}_j)$$
$$= \begin{bmatrix} 0 \\ \frac{\frac{1}{n}\sum_i h'_\gamma(\tilde{r}_i)x_{ij} + \tilde{\beta}_j \cdot \frac{1}{n}\sum_i \psi_\gamma(\tilde{r}_i)x_{ij}^2}{\lambda\alpha} \end{bmatrix}.$$

### 3.2.1 Convergence

Since SNCD fits in the general coordinate descent framework, its convergence property follows from the convergence results for coordinate descent (Tseng, 2001). To apply the results, we first show that the optimization problem is of the form

$$\min\ f(z_1, \ldots, z_m) = f_0(z_1, \ldots, z_m) + \sum_{j=1}^{m} f_j(z_j),$$

where $f_0, f_1, \ldots, f_m$ are convex, $f_0$ is first-order differentiable and the level set $\{z : f(z) \leq f(z^0)\}$ is bounded given any initial point $z^0$. A key fact to notice about this formulation is that the nondifferentiable part $\sum_j f_j(z_j)$ must be separable. The penalized Huber loss regression model in (3.4) clearly satisfies these conditions.

At each coordinate update, SNA is applied to solve the equations, which requires nonsingularity of the Newton derivative and the uniform boundedness of its inverse. When updating $\beta_0$, these requirements are met if $|\sum_i \psi_\gamma(y_i - \beta_0 - x_i^\top \beta)|$ is bounded away from 0. This is true as long as there is at least one observation with $|y_i - \beta_0 - x_i^\top \beta| \leq \gamma$. When updating $\beta_j, s_j$, it can be shown via some algebra that a sufficient condition is $0 < \alpha < 1$ and $\psi_\gamma$ is bounded. The latter always holds since $\psi_\gamma(t) \in \{0, 1/\gamma\}$ for any $t$.

In order for this local SNA strategy to work well, we also need the starting point and the optimal point in each coordinate update to be sufficiently close. Denote the globally initial $f_H$ value by $f_H^0$. Since $f_H$ decreases along SNCD iterations and the level set $\{(\beta_0, \beta) : f_H(\beta_0, \beta) \leq f_H^0\}$ is bounded, the closeness requirement is satisfied if the diameter of the set is sufficiently small.

The above discussions are summarized in the following result.

**Theorem 3.1.** *For problem* (3.4)*, let* $\lambda > 0$*,* $\alpha \in (0, 1)$ *and the initial* $f_H$ *value be* $f_H^0$*. Assume for every point* $(\beta_0, \beta)$ *in the level set* $\mathcal{L} = \{(\beta_0, \beta) : f_H(\beta_0, \beta) \leq f_H^0\}$ *there exists* $i \in \{1, \ldots, n\}$ *such that* $|y_i - \beta_0 - x_i^\top \beta| \leq \gamma$*. Then SNCD iterations converge to a global minimizer provided that the diameter of* $\mathcal{L}$ *is sufficiently small.*

To actually implement the algorithm, we still need to consider an important issue: its convergence relies on a good initial point, which is usually not guaranteed in practice. For low-dimensional problems we can use line search to ensure global convergence with an arbitrary initial point, but since line search methods involve considerable amounts of function and gradient evaluations, they are not well-suited for high-dimensional cases.

The strategy of pathwise optimization with warm start can help globalize the convergence of the algorithm. With a decreasing sequence of $\lambda$ values, this strategy sequentially solves the optimization problem at each $\lambda_k$ using the optimizer at the previous $\lambda_{k-1}$ as the initial value. When $\lambda_{k-1}$, $\lambda_k$ are reasonably close, the initial point $(\widehat{\beta}_0(\lambda_{k-1}), \widehat{\beta}(\lambda_{k-1}))$ will be near the optimizer $(\widehat{\beta}_0(\lambda_k), \widehat{\beta}(\lambda_k))$ as well. Hence each optimization problem along the path is warm-started with a good initial point, and fast convergence can be achieved. This strategy generates a solution path, which in turn will be useful for tuning parameter selection.

### 3.2.2 Comparisons

**Computational bottleneck of SNA**

Denote $\mathcal{S}(z) = (S(z_1), \ldots, S(z_p))^\top$ and $d(\beta_0, \beta) = (h'_\gamma(y_1 - \beta_0 - x_1^\top \beta), \ldots, h'_\gamma(y_n - \beta_0 - x_n^\top \beta))^\top$, then the KKT conditions (3.6) can be written compactly as

$$F(\beta_0, \beta, s) = \begin{bmatrix} -\frac{1}{n} 1^\top d(\beta_0, \beta) \\ -\frac{1}{n} X^\top d(\beta_0, \beta) + \lambda\alpha s + \lambda(1-\alpha)\beta \\ \beta - \mathcal{S}(\beta + s) \end{bmatrix} = \mathbf{0}. \tag{3.14}$$

It is easy to verify $F$ is Newton differentiable, then SNA can be directly applied here for solving $F(\beta_0, \beta, s) = \mathbf{0}$. See Appendix A for details.

In terms of computational cost, the first concern is about matrix inversion, since the Newton derivative of $F$ is a $(1+2p) \times (1+2p)$ matrix, for which inversion becomes intractable when $p$ is large. However, the decomposition (3.11) leads to an "active set strategy" that helps reduce the dimension. Given the $k$th iteration $(\beta_0^k, \beta^k, s^k)$, define the active set $A_k$ and its complement $B_k$ by

$$A_k = \{j : |\beta_j^k + s_j^k| > 1\} \text{ and } B_k = \{j : |\beta_j^k + s_j^k| \leq 1\}. \tag{3.15}$$

Then the Newton-type iteration of SNA is decomposed into two parts $A_k$ and $B_k$ and only the computation of $\beta_0^{k+1}, \beta_{A_k}^{k+1}$ requires inverting a matrix, the dimension

of which is only $(1 + |A_k|) \times (1 + |A_k|)$. In general, $|A_k|$ can be as large as $p$. But since pathwise optimization is implemented, the algorithm is warm-started at each $\lambda$ value. Hence $A_k$ is usually not too much different from the support of the optimizer, which tends to be a sparse subset of $\{1, \ldots, p\}$.

The real bottleneck is in matrix multiplication. Let $\psi_\gamma$ be as in (3.10). Let $X^* = (\mathbf{1}_n \; X)$ and $\Psi_k = \frac{1}{n}\text{diag}(\psi_\gamma(\, y_1 - \beta_0^k - x_1^\top \beta^k), \ldots, \psi_\gamma(y_n - \beta_0^k - x_n^\top \beta^k))$. Then as shown in Appendix A, each iteration includes re-computing and re-partitioning $X^{*\top}\Psi_k X^*$, which involves $O(np^2)$ arithmetic operations that become formidable for large $p$. The diagonality of $\Psi_k$ and the symmetry of $X^{*\top}\Psi_k X^*$ could be utilized to reduce computation, but the magnitude remains $O(np^2)$. Since $X^{*\top}\Psi_k X^* = \frac{1}{n}\sum_i \psi_\gamma(y - \beta_0^k - x_i^\top \beta^k) x_i^* x_i^{*\top}$, caching all the $(1 + p) \times (1 + p)$ matrices $x_i^* x_i^{*\top}$ would also speed up the computation, but since there are $n$ such matrices, such an implementation would be memory-inefficient.

**SNCD vs. SNA**

The two algorithms mainly differ in the following aspects:

- Consider a full update on $(\beta_0, \beta, s)$ as one iteration. The computational cost per iteration of SNCD is $O(np)$, compared with $O(np^2)$ for SNA.

- The SNCD iterations consist of univariate and bivariate updates only while SNA involves matrix inversions.

- While SNA has locally superlinear convergence rate in theory, SNCD is at most linear. It is a worthwhile compromise, however, considering that SNCD reduces the computational cost per iteration from $O(np^2)$ to $O(np)$ and that warm-starting due to pathwise optimization strategy allows SNCD to converge quickly.

- In practice, SNCD is much faster; and SNCD always converges while SNA diverges in some high-dimensional cases even when pathwise optimization is

used.

**SNCD vs. standard coordinate descent type algorithms**

SNCD also differs from the existing coordinate descent algorithms for penalized regression (Friedman et al., 2007; Friedman et al., 2010; Simon et al., 2011; Breheny and Huang, 2011) in the following aspects:

- It generalizes coordinate descent to work on a wider class of models where the loss functions, like the Huber loss, only need to be first-order differentiable. As shown in the next section, it is also extended to a case with a nondifferentiable loss, i.e. the quantile loss, via smoothing approximation.

- It is directly motivated from the KKT conditions as a root-finding method, where the subgradients $s_j$'s are treated as independent variables that are connected with $\beta_j$'s through the equation $\beta_j - S(\beta_j + s_j) = 0$.

- Each pair of $(\beta_j, s_j)$ is updated simultaneously with different formulas for two situations $|\tilde{\beta}_j + \tilde{s}_j| > 1$ and $|\tilde{\beta}_j + \tilde{s}_j| \leq 1$. This is quite different from the coordinate descent algorithms mentioned above that only update the coefficients $\beta_j$'s.

## 3.3 SNCD for penalized quantile regression

For the quantile loss function $\ell = \rho_\tau$, (3.1) becomes

$$\min_{\beta_0,\beta} f_Q(\beta_0, \beta) = \frac{1}{n} \sum_i \rho_\tau(y_i - \beta_0 - x_i^\top \beta) + \lambda P_\alpha(\beta). \tag{3.16}$$

SNCD cannot be directly applied to this problem since it requires the first-order derivatives of the loss function, but $\rho_\tau$ is not differentiable. However, note that

$$\rho_\tau(t) = (1-\tau)t_- + \tau t_+ = \frac{1}{2}\left\{|t| + (2\tau - 1)t\right\}.$$

Since $h_\gamma(t) \to |t|$ as $\gamma \to 0^+$, $\rho_\tau(t) \approx \frac{1}{2}\{h_\gamma(t) + (2\tau - 1)t\}$ for small $\gamma$ and the solutions to penalized quantile regression can be approximated by

$$\min_{\beta_0,\beta} f_{HA}(\beta_0, \beta) = \frac{1}{2n} \sum \left\{ h_\gamma(y_i - \beta_0 - x_i^\top \beta) + (2\tau - 1)(y_i - \beta_0 - x_i^\top \beta) \right\} + \lambda P_\alpha(\beta), \tag{3.17}$$

where "HA" stands for Huber approximation. This problem is easier to handle since its loss function is first-order differentiable. The following result provides theoretical support for this smoothing approximation.

**Theorem 3.2.** *Given any $\lambda \geq 0$, $0 < \tau < 1$ and $\{\gamma_k\}$ converging to 0, let $(\beta_{0k}, \beta_k)$ be a minimizer of $f_{HA}(\beta_0, \beta; \lambda, \tau, \gamma_k)$. Then every cluster point of sequence $\{(\beta_{0k}, \beta_k)\}$ is a minimizer of $f_Q(\beta_0, \beta; \lambda, \tau)$.*

Now we can derive the KKT conditions and apply SNCD to solve (3.17). Due to its similarity to the penalized Huber loss regression, we omit the details. At each iteration, with the current estimates denoted by $(\tilde{\beta}_0, \tilde{\beta}, \tilde{s})$ and residuals by $\tilde{r}_i$, the SNCD updates are

(i) For $\beta_0$:

$$\beta_0 \leftarrow \tilde{\beta}_0 + \frac{\sum_i \left\{ h'_\gamma(\tilde{r}_i) + 2\tau - 1 \right\}}{\sum_i \psi_\gamma(\tilde{r}_i)}.$$

(ii) For $(\beta_j, s_j)$:

(a) If $|\tilde{\beta}_j + \tilde{s}_j| > 1$, then

$$\begin{aligned} \beta_j &\leftarrow \tilde{\beta}_j + \frac{\frac{1}{2n}\sum_i \left\{ h'_\gamma(\tilde{r}_i) + 2\tau - 1 \right\} x_{ij} - \lambda\alpha \mathrm{sgn}(\tilde{\beta}_j + \tilde{s}_j) - \lambda(1-\alpha)\tilde{\beta}_j}{\frac{1}{2n}\sum_i \psi_\gamma(\tilde{r}_i) x_{ij}^2 + \lambda(1-\alpha)}, \\ s_j &\leftarrow \mathrm{sgn}(\tilde{\beta}_j + \tilde{s}_j). \end{aligned}$$

(b) If $|\tilde{\beta}_j + \tilde{s}_j| \leq 1$, then

$$\begin{aligned} \beta_j &\leftarrow 0, \\ s_j &\leftarrow \frac{\frac{1}{2n}\sum_i \left\{ h'_\gamma(\tilde{r}_i) + 2\tau - 1 \right\} x_{ij} + \tilde{\beta}_j \cdot \frac{1}{2n}\sum_i \psi_\gamma(\tilde{r}_i) x_{ij}^2}{\lambda\alpha}. \end{aligned}$$

The previous discussions on convergence and pathwise optimization also apply here. And similar to Theorem 3.1, we have the following result.

**Theorem 3.3.** *For problem* (3.16)*, let* $\lambda > 0$*,* $\alpha \in (0,1)$ *and the initial* $f_{HA}$ *value be* $f_{HA}^0$*. Assume for every point* $(\beta_0, \beta)$ *in the level set* $\mathcal{L} = \{(\beta_0, \beta) : f_{HA}(\beta_0, \beta) \leq f_{HA}^0\}$ *there exists* $i \in \{1, \ldots, n\}$ *such that* $|y_i - \beta_0 - x_i^\top \beta| \leq \gamma$ *. Then SNCD iterations converge to a global minimizer provided that the diameter of* $\mathcal{L}$ *is sufficiently small.*

### 3.3.1 The choice of $\gamma$ values

For the approximation to work well, we need to use a sufficiently small $\gamma$; but when $\gamma$ gets too close to 0, the algorithm becomes ill-conditioned. Therefore we designed a data-dependent heuristic method for picking appropriate $\gamma$ values. At each $\lambda_k$, we determine $\gamma_k$ depending on the residuals $\tilde{r}_i$'s given by the previous optimizer $(\widehat{\beta}_0(\lambda_{k-1}), \widehat{\beta}(\lambda_{k-1}))$ as follows.

i. Initialize residuals $\tilde{r}_i \leftarrow y_i$;

ii. For each $\lambda_k$:

   (a) $\gamma_k \leftarrow$ 10-th percentile of $\{|\tilde{r}_i|\}$;

   (b) $\gamma_k \leftarrow \min\{\gamma_k, \gamma_{k-1}\}$;

   (c) $\gamma_k \leftarrow \max\{\gamma_k, 0.001\}$;

   (d) solve the problem with $\gamma_k, \lambda_k$ and update $\tilde{r}_i$'s at each iteration.

In step (a) we pick a value smaller than the magnitudes of 90% of all residuals for which the loss function is the same as the quantile loss so the approximation should work well. This also keeps $\gamma_k$ above the magnitudes of 10% of the residuals, which ensures the numerical stability of the algorithm. Bracketing in (b) and (c) are additional safeguards for stability.

### 3.3.2 Related convergence results

The key to the smoothing approximation is the fact that $h_\gamma(t)$ converges to $|t|$ as $\gamma$ tends to 0. In fact, it is also easy to see that with $\gamma$ as a scaling factor, $\gamma h_\gamma(t)$ converges to the squared loss $\frac{t^2}{2}$ when $\gamma$ goes to infinity. Hence, in the same spirit of Theorem 3.2, we also show the connections between the penalized Huber loss regression and two important regression models with respectively the absolute loss and the squared loss, i.e. the Least Absolute Deviations (LAD) and the Least Squares (LS).

To simplify the notation, let $\theta = (\beta_0, \beta)$ and $P(\cdot)$ be a general penalty function. Denote

$$\begin{aligned} \min_\theta f_H(\theta; \lambda, \gamma) &= \tfrac{1}{n}\textstyle\sum_i h_\gamma(y_i - \beta_0 - x_i^\top \beta) + \lambda P(\beta), \\ \min_\theta f_A(\theta; \lambda) &= \tfrac{1}{n}\textstyle\sum_i |y_i - \beta_0 - x_i^\top \beta| + \lambda P(\beta), \\ \min_\theta f_S(\theta; \lambda) &= \tfrac{1}{2n}\textstyle\sum_i (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda P(\beta). \end{aligned}$$

Then we have $f_H(\theta; \lambda, \gamma) \to f_A(\theta; \lambda)$ as $\gamma \to 0$; $\gamma f_H(\theta; \lambda/\gamma, \gamma) \to f_S(\theta; \lambda)$ as $\gamma \to \infty$. And the following results establish the convergence between their optimizers.

**Theorem 3.4.** *Given any $\lambda \geq 0$ and $\{\gamma_k\}$ converging to 0, let $\theta_k$ be a minimizer of $f_H(\theta; \lambda, \gamma_k)$. Then every cluster point of sequence $\{\theta_k\}$ is a minimizer of $f_A(\theta; \lambda)$.*

**Theorem 3.5.** *Given any $\lambda \geq 0$ and $\{\gamma_k\}$ converging to $\infty$, let $\theta_k$ be a minimizer of $f_H(\theta; \lambda/\gamma_k, \gamma_k)$. Then every cluster point of sequence $\{\theta_k\}$ is a minimizer of $f_S(\theta; \lambda)$.*

Therefore, the penalized Huber loss regression bridges the gap between LAD and LS regression as $\gamma$ varies from 0 to $\infty$. The solutions of the penalized Huber loss regression constitute a rich spectrum from the solution of LAD regression to that

of LS regression. This property gives us more flexibility in fitting high-dimensional regression models.

### 3.4 Adaptive strong rule for screening predictors

Tibshirani et al. (2012) proposed the (sequential) strong rule for screening out predictors in pathwise optimization of penalized regression models for computational efficiency. However, when applied to the penalized Huber loss regression and quantile regression, we discover that the strong rule suffers from the issue of "violations" that is explained below. To deal with this issue and enhance algorithmic stability, we develop an adaptive version of the strong rule.

We first describe the strong rule. Consider a general elastic-net penalized regression

$$\min_{\beta_0,\beta} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i - \beta_0 - x_i^\top \beta) + \lambda P_\alpha(\beta).$$

where $\ell$ is convex and differentiable. Then the optimizer $(\widehat{\beta}_0(\lambda), \widehat{\beta}(\lambda))$ satisfies the KKT conditions

$$\begin{cases} -\frac{1}{n}\sum_i \ell'(y_i - \widehat{\beta}_0 - x_i^\top \widehat{\beta}) = 0, \\ -\frac{1}{n}\sum_i \ell'(y_i - \widehat{\beta}_0 - x_i^\top \widehat{\beta}) x_{ij} + \lambda\alpha\widehat{s}_j + \lambda(1-\alpha)\widehat{\beta}_j = 0, \\ \widehat{s}_j \in \partial|\widehat{\beta}_j|, \quad j = 1, \ldots, p. \end{cases}$$

The unpenalized intercept $\beta_0$ is always in the model, so there is no screening rule for it. For $\beta_j$, let $c_j(\lambda) = -\frac{1}{n}\sum_i \ell'(y_i - \widehat{\beta}_0 - x_i^\top \widehat{\beta}) x_{ij}$. Assume each $c_j$ is $\alpha$-Lipschitz continuous,

$$|c_j(\lambda) - c_j(\lambda')| \le \alpha|\lambda - \lambda'|, \quad \text{for every} \quad \lambda, \lambda' > 0. \tag{3.18}$$

Then at each new $\lambda_k$ in the solution path, given the previous optimizer $(\widehat{\beta}_0(\lambda_{k-1}), \widehat{\beta}(\lambda_{k-1}))$ and the corresponding $c_j(\lambda_{k-1})$'s, the strong rule discards predictor $j$ if

$$|c_j(\lambda_{k-1})| < \alpha(2\lambda_k - \lambda_{k-1}). \tag{3.19}$$

The reasoning is as follows. Assume (3.18) and (3.19) hold, since $\lambda_{k-1} > \lambda_k$, we have

$$\begin{aligned} |c_j(\lambda_k)| &\leq |c_j(\lambda_k) - c_j(\lambda_{k-1})| + |c_j(\lambda_{k-1})| \\ &< \alpha(\lambda_{k-1} - \lambda_k) + \alpha(2\lambda_k - \lambda_{k-1}) \\ &= \alpha\lambda_k. \end{aligned}$$

It follows that $\widehat{\beta}_j(\lambda_k) = 0$, since by contradiction $\widehat{\beta}_j(\lambda_k) \neq 0$ implies $\widehat{s}_j(\lambda_k) = \text{sgn}(\widehat{\beta}_j(\lambda_k))$ thus $|c_j(\lambda_k)| = \lambda_k\alpha + \lambda_k(1-\alpha)|\widehat{\beta}_j(\lambda_k)| \geq \lambda_k\alpha$.

The effectiveness of the strong rule relies on the assumption (3.18), which does not necessarily hold. So application of the rule should always be accompanied with a check of the KKT conditions. A pathwise optimization algorithm incorporating the strong rule proceeds as follows.

For each $\lambda_k$,

(a) Compute the eligible set $E = \{j : |c_j(\lambda_{k-1})| \geq \alpha(2\lambda_k - \lambda_{k-1})\}$;

(b) Solve the problem using only the predictors in $E$;

(c) Check KKT conditions on the solution: $|c_j(\lambda_k)| \leq \alpha\lambda_k$ for $j \in E^c$. We are done if there are no violations; otherwise, add violating indices to $E$ and repeat (b) and (c).

For the penalized least squares and logistic regression we have not encountered any violation, but it a different story for the penalized Huber loss regression and quantile regression. Using the strong rule for these two models, we often encounter a large number of violations, indicating that the rule may have been too restrictive. Since the algorithm is re-run each time violations are found, the overall efficiency is affected. Thus reducing the number of violations can enhance the algorithmic stability and lead to potential speedup.

A simple approach is to use a multiplier $M > 1$ and relax the assumption (3.18) to the following: $\forall \lambda, \lambda' > 0$,

$$|c_j(\lambda) - c_j(\lambda')| \leq \alpha M |\lambda - \lambda'|.$$

Accordingly, we will need to change (3.19) to

$$|c_j(\lambda_{k-1})| < \alpha \left(\lambda_k + M(\lambda_k - \lambda_{k-1})\right).$$

However, this strategy does not work well in practice, since it is difficult to pre-determine an appropriate value of $M$ that suits all values of $\lambda$ in the solution path.

Hence we propose an "adaptive" version that allows $M$ to vary with $\lambda$. This rule automatically estimates a localized $M(\lambda)$ that varies and adapts to the trends of the solution paths, which reduces the number of violations by a large margin without sacrificing speed. The idea is as follows.

Let $M(\lambda_0) = 1$. Then at each $\lambda_k$,

(a) use $M(\lambda_{k-1})$ to construct the eligible set, i.e. let

$$E = \{j : \ |c_j(\lambda_{k-1})| \geq \alpha \left(\lambda_k + M(\lambda_{k-1})(\lambda_k - \lambda_{k-1})\right)\};$$

(b) solve the problem using only the predictors in $E$, and check KKT conditions as before; update $E$ and repeat step (b) if violations occur;

(c) compute $M(\lambda_k)$ based on the local trend of $c_j$'s:

$$M(\lambda_k) = \frac{\max\limits_{1 \leq j \leq p} |c_j(\lambda_{k-1}) - c_j(\lambda_k)|}{\alpha(\lambda_{k-1} - \lambda_k)}.$$

## 3.5 Optimization performance for penalized quantile regression

As mentioned in the introduction, `quantreg` is another publicly available R package that supports lasso penalized quantile regression. Since our implementation employs an approximation model, it does not give "exact" solutions. Hence we want to compare its solutions with the ones computed by `quantreg` in terms of optimality.

Unlike `hqreg` that computes a solution path, `quantreg` computes a single solution for a given $\lambda$ value, and it does not support the general elastic-net penalty with $0 < \alpha < 1$. For comparison, we only consider lasso ($\alpha = 1$). We first computed a solution path along 100 $\lambda$ values using `hqreg` and then ran `quantreg` for each $\lambda$ value. Note that `quantreg` actually uses the formulation

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \rho_\tau(y_i - \beta_0 - x_i^\top \beta) + \lambda \cdot \frac{1}{2} \sum_{j=1}^{p} |\beta_j|$$

which does not have a $1/n$ scaling factor for the loss part and instead contains a $1/2$ factor for the penalty. This is intended to treat the penalty terms as if median regression were performed on them ($\frac{1}{2}\lambda|\beta_j| = \rho_{0.5}(\lambda\beta_j)$). Due to this difference, for each $\lambda$ value used with `hqreg`, we equivalently supplied `quantreg` with $2n\lambda$. Also, while `hqreg` supports data preprocessing via the argument "preprocess" with 3 options "standardize", "rescale" and "none", `quantreg` does not provide such an option. So we standardized the data beforehand for all the real datasets involved in this section and used the standardized ones for comparison. Consequently, we set `preprocess = "none"` when calling `hqreg`. For `quantreg`, the latest version 5.24 was used.

Let $f_Q(\cdot;\lambda)$ denote the objective function as in (3.16), and let $\widehat{\beta}_{\text{hqreg}}$ and $\widehat{\beta}_{\text{quantreg}}$ be the solutions given by the two packages, respectively. For $\alpha = 1$ the model is not strictly convex, so in general it does not have a unique optimizer. Hence the values of the two solutions may not be very close. Instead, a reasonable approach

is to compare the values of the objective functions $f_Q(\widehat{\beta}_{\text{hqreg}})$ and $f_Q(\widehat{\beta}_{\text{quantreg}})$. Specifically, we made the comparisons based on the relative difference,

$$D(\lambda) = \frac{f_Q(\widehat{\beta}_{\text{hqreg}}; \lambda) - f_Q(\widehat{\beta}_{\text{quantreg}}; \lambda)}{f_Q(\widehat{\beta}_{\text{quantreg}}; \lambda)}. \tag{3.20}$$

Two datasets were considered:

- GDP (Koenker and Machado, 1999): consists of 161 observations on national GDP growth rates, recorded as "Annual Change Per Capita GDP", and 13 covariates. The first 71 observations are from the period 1965-1975, and the rest from the period 1975-1985. This dataset is available in `quantreg` via `data(barro)`.

- Riboflavin (Bühlmann et al., 2014): gene-expression data for predicting log transformed riboflavin (vitamin B2) production rate in Bacillus subtilis. It contains 71 observations and 4088 features (gene expressions). This dataset is available in R package `hdi` via `data(riboflavin)`. For this task only 1000 features with the largest variances were used.

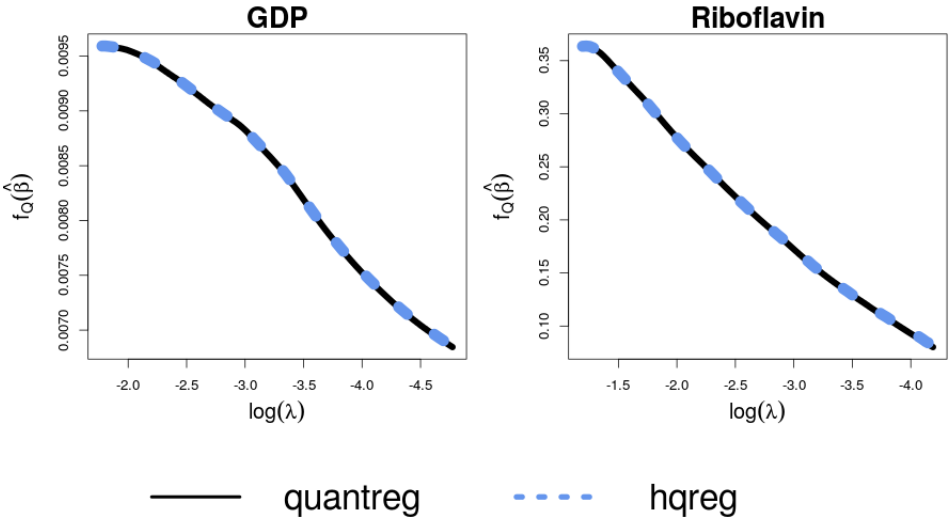


Figure 3.1: Values of objective functions with $\tau = 0.5$ along the solution path for GDP and riboflavin datasets. Solid line: `quantreg`, dashed line: `hqreg`.

| Dataset | $\tau$ | $\min D(\lambda_i)$ | $\max D(\lambda_i)$ |
|---|---|---|---|
| GDP | 0.25 | -2.1e-9 | 1.5e-3 |
| | 0.50 | -1.3e-10 | 9.6e-4 |
| | 0.75 | -3.0e-10 | 1.7e-3 |
| Riboflavin | 0.25 | 6.5e-5 | 2.6e-2 |
| | 0.50 | -3.6e-10 | 2.0e-2 |
| | 0.75 | 8.8e-5 | 2.1e-2 |

Table 3.1: Range of $D(\lambda_i), 1 \leq i \leq 100$

| Dataset | $\tau$ | `hqreg` | `quantreg` | | |
|---|---|---|---|---|---|
| | | | total | $\lambda_1$ | $\lambda_{100}$ |
| GDP | 0.25 | 0.018 | 0.235 | 0.007 | 0.002 |
| | 0.50 | 0.018 | 0.223 | 0.002 | 0.003 |
| | 0.75 | 0.027 | 0.240 | 0.003 | 0.002 |
| Riboflavin | 0.25 | 2.501 | 538.2 | 3.630 | 4.958 |
| | 0.50 | 3.026 | 531.6 | 4.591 | 4.984 |
| | 0.75 | 2.922 | 588.8 | 7.119 | 5.791 |

Table 3.2: Running time (in seconds) for computing the solution paths

Figure 3.1 displays the computed values of objective functions $f_Q(\widehat{\beta}_{\text{hqreg}})$ and $f_Q(\widehat{\beta}_{\text{quantreg}})$ for $\tau = 0.5$ along the solution path for both datasets. There is no visually detectable discrepancy between the two lines. Hence we also computed the range of $D(\lambda)$ in each case and the results are listed in Table 3.1. In each case, the range of $D(\lambda_i)$'s is extremely narrow and all values are very close to zero. This indicates the two packages indeed have similar performances.

We also report the running time in Table 3.2. The time for `hqreg` is for one call that fits the entire solution path, and the time for `quantreg` is the total of time recorded separately for each $\lambda$. For all these cases `hqreg` is significantly faster than `quantreg`, although it may not be quite fair for `quantreg` since it does not rely on warm-start. The timings taken for `quantreg` on $\lambda_1$ and $\lambda_{100}$ are also listed, which appear to be roughly the same. In the case of riboflavin data, the running time of `quantreg` on single $\lambda$ values is in fact longer than the time used by `hqreg` to compute the whole path.

To further investigate their performances in various other scenarios, we ran a large set of experiments on 10000 datasets, each generated with the following settings:

- the number of observations $n$ and the number of features $p$ are randomly selected from the set $\{20, 100, 200, 500, 1000, 2000, 5000\}$.

- the number of nonzero coefficients is $q = \theta \min(n, p)$ where $\theta$ is uniformly sampled from $\{5\%, 10\%, 20\%, 30\%\}$ and the coefficients values are randomly selected from $\{\pm 1, \ldots, \pm 10\}$.

- each feature vector $x_i$ is generated via $x_{ij} = z_{ij} + 0.5u_i,\ 1 \le j \le p$, where $z_{ij}, u_i$'s are i.i.d. standard gaussian, so that each pair of features has the same correlation 0.25.

- the outcome $y_i$'s are generated by $y_i = 10 + x_i^\top \beta + \varepsilon_i$, where $\varepsilon_i$'s are iid sampled

from Student's t distribution with $df = 4$.

For each dataset and each $\tau \in \{0.25, 0.5, 0.75\}$, we applied `hqreg` to compute an entire solution path and randomly selected an index $k$ out of $\{10, 20, \ldots, 100\}$, then ran `quantreg` on $\lambda_k$, the k-th $\lambda$ value for the solution path computed by `hqreg`. These experiments were performed in parallel via grid computing on a high performance cluster at the University of Iowa.

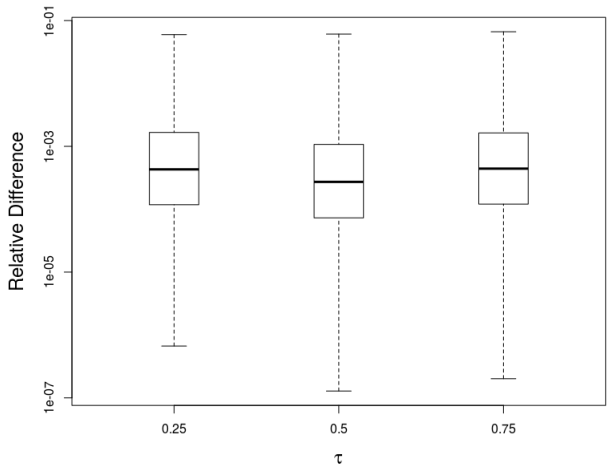


Figure 3.2: Boxplots of the relative difference $D$ on 10000 simulated datasets

We calculate the relative difference $D(\lambda)$ for each pair of the solutions and summarize the results in three boxplots plotted on the logarithmic scale shown in Figure 3.2. We observe that the values of $D$ have a narrow range between 1e-7 and 1e-1 with the majority falling below 1e-3, and are slightly smaller for $\tau = 0.5$. Besides, the distribution of $D$ appears roughly symmetric on the logarithmic scale in each case.

## 3.6 Timing performance

In addition to the Huber loss and the quantile loss, `hqreg` also supports the squared loss for the least squares which is not discussed here, but its SNCD iterations can be derived in a similar way as the other two models. Here we consider their

running time performances.

We generated Gaussian data with $n$ observations and $p$ features, where each pair of features have an identical correlation $\rho$. To simplify settings and highlight the timing comparison based on the key parameter $\gamma$ and $\tau$, we set $\rho = 0.25$ and $\alpha = 0.9$ for all cases. The responses were generated by

$$y = \sum_j x_j \beta_j + k \cdot \varepsilon$$

where $\beta_j = (-1)^j \exp(-(j-1)/10)$, $\varepsilon \sim T(df = 4)$ and $k$ is determined so that the signal-to-noise ratio is 3.

Huber Loss Regression and Least Squares

In this part, we compare the running time of competing methods for the elastic-net penalized Huber loss regression and least squares. For the Huber loss, since there is no other algorithms, we consider only `hqreg` for SNCD with no variable screening (`hqreg`-NVS), SNCD with the adaptive strong rule (`hqreg`-ASR), and our implementation of pure SNA. In the experiments we considered 5 values of $\gamma$ ranging from 0.01 to 100. On the other hand, for the least squares we compared `hqreg` with a state-of-the-art coordinate descent algorithm implemented by R package `glmnet`. For `glmnet`, the latest version 2.0-5 was used which employs the strong rule for variable screening. All methods considered here are R functions. `glmnet` does all its computation in Fortran, `hqreg` does the computation in C, and the SNA implementation is also programmed in C with matrix operations performed via BLAS and LAPACK.

We have found in practice that convergence of SNA has much higher reliance on initial points than SNCD, and it can fail if the $\lambda$ sequence is not dense enough. Hence we divided the experiments into two parts. In the first part for the Huber loss and the least squares together, we left out SNA and computed each solution

| | Huber | | | | | Least Squares |
|---|---|---|---|---|---|---|
| | | | $\gamma$ | | | |
| | 0.01 | 0.1 | 1 | 10 | 100 | |
| | | | $n = 1000, p = 100$ | | | |
| hqreg-NVS | 0.61 | 0.09 | 0.05 | 0.04 | 0.04 | 0.03 |
| hqreg-ASR | 0.33 | 0.06 | 0.03 | 0.02 | 0.02 | 0.02 |
| glmnet | — | — | — | — | — | 0.02 |
| | | | $n = 5000, p = 100$ | | | |
| hqreg-NVS | 2.46 | 0.48 | 0.24 | 0.20 | 0.21 | 0.14 |
| hqreg-ASR | 1.32 | 0.30 | 0.15 | 0.12 | 0.11 | 0.07 |
| glmnet | — | — | — | — | — | 0.02 |
| | | | $n = 100, p = 1000$ | | | |
| hqreg-NVS | 1.89 | 0.39 | 0.09 | 0.08 | 0.08 | 0.05 |
| hqreg-ASR | 0.52 | 0.11 | 0.03 | 0.02 | 0.02 | 0.02 |
| glmnet | — | — | — | — | — | 0.02 |
| | | | $n = 100, p = 5000$ | | | |
| hqreg-NVS | 8.70 | 2.09 | 0.46 | 0.38 | 0.38 | 0.29 |
| hqreg-ASR | 0.85 | 0.23 | 0.09 | 0.11 | 0.08 | 0.07 |
| glmnet | — | — | — | — | — | 0.08 |
| | | | $n = 100, p = 20000$ | | | |
| hqreg-NVS | 30.27 | 8.88 | 2.43 | 2.45 | 2.40 | 1.23 |
| hqreg-ASR | 1.60 | 0.54 | 0.43 | 0.31 | 0.30 | 0.32 |
| glmnet | — | — | — | — | — | 0.30 |
| | | | $n = 100, p = 100000$ | | | |
| hqreg-NVS | 175.81 | 45.33 | 11.23 | 11.49 | 11.41 | 5.94 |
| hqreg-ASR | 4.50 | 2.12 | 1.69 | 1.58 | 1.53 | 1.57 |
| glmnet | — | — | — | — | — | 1.39 |

Table 3.3: Running time (in seconds) for computing regularization paths for the elastic-net penalized Huber loss regression and least squares regression. Total time for `100` $\lambda$ values, averaged over 3 runs.

path with the usual number of 100 $\lambda$ values. In the second part, we compared only SNA and SNCD(`hqreg`-NVS) for the Huber loss on dense lambda sequences each consisting of 10000 values.

| | $\gamma$ | | |
|---|---|---|---|
| | 0.1 | 1 | 10 |
| | $n = 1000, p = 100$ | | |
| SNA | 3.98 | 5.16 | 5.44 |
| SNCD | 1.89 | 1.27 | 1.19 |
| | $n = 5000, p = 100$ | | |
| SNA | 17.98 | 24.42 | 26.33 |
| SNCD | 12.38 | 6.84 | 6.31 |
| | $n = 100, p = 1000$ | | |
| SNA | $\times$ | 11.70 | 10.47 |
| SNCD | 2.24 | 1.62 | 1.51 |
| | $n = 100, p = 5000$ | | |
| SNA | $\times$ | 98.66 | 100.76 |
| SNCD | 9.87 | 8.19 | 8.96 |

Table 3.4: Running time (in seconds) for comparing SNCD(`hqreg`-NVS) and SNA on the penalized Huber loss regression. "$\times$" represents early exit due to divergence at some $\lambda$ value. Total time for `10000` $\lambda$ values.

Table 3.3 shows average CPU timings for the first part. First compare the timings for the Huber loss. Across different values of $\gamma$, we observe that for both versions the timings increase when $\gamma$ is nearing 0, and stay almost the same for

$\gamma \geq 1$. And clearly `hqreg`-ASR that employs the adaptive strong rule is much faster and more scalable than `hqreg`-NVS that has no variable screening. For the least squares, `hqreg`-ASR and `glmnet` have similar performances except the case with $n = 5000$. Besides, we discover that the timings for the Huber loss regression with $\gamma \geq 1$ are very close to those of the least squares. Considering that the Huber loss is more difficult to handle than the simple squared loss, the performance of `hqreg` is very impressive.

Table 3.4 shows average CPU timings for the second part. We observe that while SNCD converges in every case, SNA fails in the cases with large $p$ and $\gamma = 0.1$. When $p$ is small, SNCD does not have much advantage. But when $p$ increases, SNCD becomes considerably faster with an increasing speedup relative to SNA. These results show that SNCD is more stable and scalable than SNA.

Quantile Regression

`hqreg` is faster than `quantreg` for the examples in section 3.5. However, `quantreg` does not implement pathwise optimization and rely on warm-start like `hqreg` does. Instead, for each supplied $\lambda$ value it has to solve the corresponding problem individually "from scratch". So it is not quite reasonable to compare `quantreg` with `hqreg` for computing the whole solution path. For this part, we compare only `hqreg`-NVS and `hqreg`-ASR. As shown in Table 3.5, `hqreg`-ASR is similar to `hqreg`-NVS in cases with $p = 100$ but considerably faster when $p$ gets larger. `hqreg`-ASR also shows much better scalability with the dimension $p$.

## 3.7 Breast cancer gene expression data example

We now compare the modelling performance of penalized Huber loss regression, quantile regression and least squares via an empirical analysis on a real dataset. It is a breast cancer gene expressions dataset that comes from the Cancer Genome

| | $\tau$ | | |
|---|---|---|---|
| | 0.25 | 0.50 | 0.75 |
| | $n = 1000, p = 100$ | | |
| hqreg-NVS | 0.21 | 0.18 | 0.19 |
| hqreg-ASR | 0.13 | 0.10 | 0.11 |
| | $n = 5000, p = 100$ | | |
| hqreg-NVS | 0.56 | 0.58 | 0.54 |
| hqreg-ASR | 0.38 | 0.42 | 0.33 |
| | $n = 100, p = 1000$ | | |
| hqreg-NVS | 10.77 | 7.37 | 11.90 |
| hqreg-ASR | 2.98 | 1.94 | 2.92 |
| | $n = 100, p = 5000$ | | |
| hqreg-NVS | 47.08 | 41.46 | 58.97 |
| hqreg-ASR | 3.33 | 2.92 | 4.23 |
| | $n = 100, p = 100000$ | | |
| hqreg-ASR | 19.28 | 12.43 | 22.98 |

Table 3.5: Running time (in seconds) for computing regularization paths for penalized quantile regression. Total time for `100` $\lambda$ values, averaged over 3 runs.

Atlas (2012) project (`http://cancergenome.nih.gov/`), obtained using Agilent mRNA expression microarrays. It contains expression measurements of 17814 genes on 536 patients, including BRCA1, the first gene identified to be associated with increasing risk of early onset breast cancer. Hence we regress the key gene BRCA1

on the other genes to detect potential interconnections. Before fitting the models, we carried out the following two screening steps: remove any gene for which the range of the expression among all patients is less than 2, and remove any gene for which the sample correlation with BRCA1 is less than 0.05. After the screening, there are 11562 genes left.

Let IQR($y$) be the inter-quartile range of $y$. We consider 7 elastic-net penalized linear regression models using these genes as predictors: the least squares (LS-Enet); 3 Huber loss regression models with $\gamma$ taking the values IQR($y$), IQR($y$)/2, IQR($y$)/10, where IQR($y$) = 0.93, denoted as H-Enet($\gamma$ = IQR($y$)), H-Enet($\gamma$ = IQR($y$)/2), and (H-Enet($\gamma$ = IQR($y$)/10), respectively; 3 quantile regression models with $\tau = 0.25, 0.50, 0.75$, denoted as Q-Enet($\tau = 0.25$), Q-Enet($\tau = 0.50$), and Q-Enet($\tau = 0.75$), respectively. We use $\alpha = 0.9$ in the elastic-net penalty for all the models.

We conduct 50 random partitions. For each partition, we randomly select 300 patients as the training data and the other 236 as the testing data. A five-fold cross validation is applied to the training data to select the tuning parameter $\lambda$. For prediction on the testing set, we consider two error measures. The first one is the commonly used mean absolute prediction error (MAPE). Since MAPE is not sensitive to heterogeneity and may not provide accurate assessment for Q-Enet($\tau = 0.25$) and Q-Enet($\tau = 0.75$) which use asymmetric losses, we also consider using the quantile loss $\rho_\tau$ to measure prediction performance as suggested in Wang et al. (2012). With $\rho_\tau$ for corresponding quantile regression models and $\rho_{0.5}$ for the least squares and the Huber loss regression models, we define quantile-based prediction error (QPE) as $\sum_i \rho_\tau(y_i - \widehat{y}_i)/n$.

In Table 3.6 we report the average number of nonzero regression coefficients, the average MAPE and the average QPE, where numbers in the parentheses are the corresponding standard errors across the 50 partitions. The standard errors of the

| Model | Ave # nonzero | Ave MAPE | Ave QPE |
|---|---|---|---|
| LS-Enet | 114.30 (36.99) | 0.335 (0.018) | 0.167 (0.009) |
| H-Enet($\gamma = \text{IQR}(y)$) | 100.14 (44.70) | 0.331 (0.018) | 0.166 (0.009) |
| H-Enet($\gamma = \text{IQR}(y)/2$) | 82.06 (30.40) | 0.310 (0.020) | 0.155 (0.010) |
| H-Enet($\gamma = \text{IQR}(y)/10$) | 114.08 (30.73) | 0.293 (0.021) | 0.146 (0.010) |
| Q-Enet($\tau = 0.25$) | 94.58 (41.60) | 0.373 (0.026) | 0.151 (0.010) |
| Q-Enet($\tau = 0.50$) | 152.90 (51.96) | 0.294 (0.021) | 0.147 (0.012) |
| Q-Enet($\tau = 0.75$) | 104.90 (27.96) | 0.317 (0.027) | 0.109 (0.007) |

Table 3.6: Analysis of the microarray dataset

estimated numbers of nonzero coefficients are large relative to the averages, showing that all models are affected by noise to some extent. However, the standard errors for MAPE and QPE are relatively small, which indicates the prediction performances are stable. Among all models, H-Enet($\gamma = \text{IQR}(y)/10$) and Q-Enet($\tau = 0.50$) have the best performances in terms of MAPE, and Q-Enet($\tau = 0.75$) dominates QPE, while LS-Enet performs poorly under both criteria. Q-Enet($\tau = 0.75$) seems the best overall and it also tends to select sparser models compared to the aforementioned H-Enet($\gamma = \text{IQR}(y)/10$), Q-Enet($\tau = 0.50$) or LS-Enet.

For each model, different partitions may lead to different selection results. We select LS-Enet, H-Enet($\gamma = \text{IQR}(y)/10$) and Q-Enet($\tau = 0.75$) to represent their own classes, and report the names and the frequencies of top genes selected (over 40 times) in Table 3.7 where the genes are ordered alphabetically. We observe that some genes such as DTL, NBR2, PSME3, RPL27 have high frequencies with all three models, while genes such as KHDRBS1 do not. Overall, H-Enet($\gamma = \text{IQR}(y)/10$) and Q-Enet($\tau = 0.75$) select more genes with high frequencies than LS-Enet while their model sizes are smaller on average, especially Q-Enet($\tau = 0.75$). It indicates these two models more consistently capture the important genes.

| LS-Enet | | H-Enet($\gamma = \text{IQR}(y)/10$) | | Q-Enet($\tau = 0.75$) | |
|---|---|---|---|---|---|
| Gene | Frequency | Gene | Frequency | Gene | Frequency |
| DTL | 45 | C17orf53 | 46 | C17orf53 | 48 |
| KHDRBS1 | 41 | CENPQ | 42 | CENPM | 45 |
| NBR2 | 50 | DTL | 46 | DTL | 44 |
| PSME3 | 45 | MCM6 | 50 | GCN5L2 | 44 |
| RPL27 | 45 | NBR1 | 47 | KIAA0101 | 40 |
| VPS25 | 43 | NBR2 | 50 | MCM6 | 42 |
| | | NMT1 | 41 | NBR1 | 49 |
| | | PSME3 | 50 | NBR2 | 50 |
| | | RPL27 | 41 | PSME3 | 50 |
| | | | | RPL27 | 50 |
| | | | | SUZ12 | 40 |
| | | | | SYNGR4 | 41 |
| | | | | XRCC2 | 41 |

Table 3.7: Genes selected with high frequency for the microarray dataset

# CHAPTER 4
# HETEROGENEITY DISCOVERY REGRESSION (HDR)

## 4.1 Introduction

Heterogeneity is a pervasive problem in statistics. For example, in clinical trials it is a common phenomenon that a certain target treatment can have different effects on different subjects. This kind of heterogeneity may be driven by unobserved latent factors, or unknown interactions between factors. Although data heterogeneity often go unnoticed, they may have serious effects on statistical estimation and inference.

A popular method for analyzing heterogeneous data is to treat data as sampled from a mixture of subgroups with separate sets of parameters and then apply finite mixture model analysis (Everitt, 1981). The mixture model approach requires specification of underlying distribution and the number of mixture components which are both difficult problems in their own rights. More importantly, this approach works only when the subgroups are balanced or at least each subgroup has enough mass. This may not be true for many real data where there are only small heterogeneous subgroups or individual outliers that do not form subgroups at all.

In the context of linear regression, robust regression methods were proposed to provide presumably trustworthy coefficient estimation in the presence of heterogeneity or outliers. In the literature, the breakdown point (Hampel, 1971) of an estimator is a popular measure of robustness. Intuitively, it is the maximum proportion of heterogeneity a method can handle before giving largely biased estimates. Huber's M estimator (Huber (1981), hereby referred to as Huber loss regression) is one of the most efficient and widely used robust regression methods, yet it has a breakdown point of 0. On the other hand, methods with a high breakdown point, such as S-estimator (Rousseeuw and Yohai, 1984), Least Median of Squares

(Rousseeuw, 1984) and MM-estimator (Yohai, 1987), typically have computational costs that grow exponentially in the dimension. In addition, it is not straightforward to apply these robust regression methods to the task of heterogeneity detection.

We propose a new approach to automatically detect heterogeneity and provide accurate coefficient estimation for linear regression in a potentially high-dimensional setting. Let $\{(y_i, x_i) \in \mathbb{R} \times \mathbb{R}^p : i = 1, \ldots, n\}$ be the observed sample. After adjusting for the homogeneous effects of the covariates, we model the heterogeneity through an extra variable $d_i$ and its subject-specific coefficient $\tau_i$:

$$y_i = d_i\tau_i + x_i^\tau \beta + \varepsilon_i, \quad i = 1, \ldots, n, \tag{4.1}$$

where $\beta$ is the vector of coefficients (including an intercept) common across all subjects, $\varepsilon_i$ are the error terms independent of $d_i$ and $x_i$, and $\tau_i$'s are the subject-specific "deviation effects". $d_i$ here is intended to incorporate two scenarios:

- $d_i = 1$ when the source of heterogeneity is unknown or unobserved;
- $d_i$ is a component of $x_i$, say $d_i = x_{ij}$ for some $j$, when the heterogeneity is known to be driven by the observed $x_{ij}$'s.

As for $\tau$, rather than taking a rigid, parametric view on its nature, we only assume that it is inherently sparse in the sense that only a small proportion of $\tau_i$'s are nonzero with no restrictions on what values they may take.

The rest of this chapter is organized as follows. In Section 4.2, we describe the proposed approach Heterogeneity Discovery Regression (HDR) and its assumptions in details. In Section 4.3, we establish the theoretical properties of HDR, first a deterministic theorem for generic loss functions, followed by a probabilistic corollary for least squares loss that guarantees statistical precision for any local optimum of the composite objective with high probability. Section 4.4 discusses the computational aspect of HDR. In Section 4.5, we investigate and compare the empirical performances of HDR with competitive methods through simulation studies.

Section 4.6 illustrates the application of HDR to a real data example.

Notations: For real vector $v \in \mathcal{R}^m$, we denote its $p$-norm by $\|v\|_p$ such that $\|v\|_p = (\sum_{i=1}^m |v_i|^p)^{1/p}$ for $1 \leq p < \infty$, $\|v\|_\infty = \max_{1\leq i\leq m} |v_i|$ and $\|v\|_0$ represents the number of nonzero elements in $v$. For a function $f : \mathcal{R}^m \mapsto \mathcal{R}$ and any $z \in \mathrm{dom} f$, if $f$ is differentiable at $z$ we use $\nabla f(z)$ to denote the gradient; otherwise, if $f$ is locally Lipschitz continuous at $z$, we use $\partial f(z)$ to denote the (generalized) subdifferential as defined in Chapter 2. For two functions $f(n)$ and $g(n)$, we write $f(n) \gtrsim g(n)$ if $f(n) \geq cg(n)$ for some $c \in (0, \infty)$ that does not depend on $n$, and $f(n) \gg g(n)$ or $g(n) = o(f(n))$ to mean $g(n)/f(n) \to 0$ as $n \to \infty$. The notations $f(n) \lesssim g(n)$ and $f(n) \ll g(n)$ are defined similarly in an opposite direction.

## 4.2 Formulation

Denote $D = \mathrm{diag}(d_1, \ldots, d_n)$, $X = (x_1, \ldots, x_n)^\top$, $y = (y_1, \ldots, y_n)^\top$. Then (4.1) can be expressed in matrix form as

$$y = D\tau + X\beta + \varepsilon.$$

Ignoring the presence of the heterogeneity component $D\tau$, the most commonly used ordinary least squares estimate for model (4.1) will be

$$\widehat{\beta}_{OLS} = (X^\top X)^{-1} X^\top y = \beta + (X^\top X)^{-1} X^\top (D\tau + \varepsilon),$$

which has a conditional bias $E(\widehat{\beta}_{OLS} - \beta | X, \beta, D, \tau) = (X^\top X)^{-1} X^\top D\tau$, linear in $D\tau$. So this estimate can be highly distorted even when only one $d_i\tau_i$ is nonzero but relatively large. Consequently, any statistical inference based on least squares is highly unreliable.

A natural way to fix this problem is to include $D\tau$ in the loss, with the specification of $D$ as described in the previous section. With this inclusion, we are faced with a situation with $n + p$ unknown coefficients which is greater than

the sample size $n$. The ordinary least squares estimator ceases to be useful in this high-dimensional case due to model nonidenfiability and computational infeasibility.

We consider penalized regression to deal with these problems. Denote $\theta = (\tau^\top, \beta^\top)^\top$, then our proposed method, Heterogeneity Discovery Regression (HDR), is to simultaneously estimate $\beta$ and detect heterogeneity by solving

$$\min_{\|\theta\|_1 \leq R} \mathcal{L}_n(\theta) + \rho_\lambda(\theta), \tag{4.2}$$

where $\mathcal{L}_n(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i - d_i\tau_i - x_i^\top\beta)$ with $\ell$ being a generic loss function for the residuals, and $\rho_\lambda(\theta)$ a regularizer that induces sparsity in the solution. In particular, $\rho_\lambda(\theta)$ need to be separable across the components of $\theta$, and it also allows different levels of penalization on the two parts $\tau$ and $\beta$. In other word, by slightly abusing the notation, we can write

$$\rho_\lambda(\theta) = \sum_{i=1}^n \rho_{\lambda_1}(\tau_i) + \sum_{j=1}^p \rho_{\lambda_2}(\beta_j), \tag{4.3}$$

where $0 < \lambda_1 \leq \lambda$ and $0 \leq \lambda_2 \leq \lambda$. In our theory we allow the penalties $\rho_{\lambda_1}$, $\rho_{\lambda_2}$ to be nonconvex. Due to this potential nonconvexity, our proposed approach also includes a side constraint $\|\theta\|_1 \leq R$. As long as the composite objective (4.2) is continuous, the Weierstrass extreme value theorem guarantees the existence of a global minimum $\widehat{\theta}$ within the constrained region. She and Owen (2012) considered a similar penalized formulation for outlier detection, and proposed an algorithm called $\Theta$-IPOD. Compared with their paper, a main contribution of this work is establishment of the statistical properties of the estimates (any local minimizer) that apply to a wide class of loss functions and penalties, and these results can be extended to the case that $\tau_i$'s are vectors for which group penalties are considered.

### 4.2.1 Theoretical challenges

Consider the simpler case where $p \ll n$, then we may remove the penalization on $\beta$. Since we do not know which $\tau_i$'s are nonzero, we still need to penalize $\tau$ in HDR, otherwise the model will be nonidentifiable. Now imagine an unrealistic situation such that we do know which observations are heterogeneous with $\tau_i \neq 0$, then there is no need for penalization and the problem is much simplified. A natural question arises: how well can we estimate $\beta$ and the nonzero $\tau_i$'s in this ideal situation?

In such scenario, let $\tau^*$ and $\beta^*$ denote the true values, and let $S = \{i : \tau_i^* \neq 0\}$ be the support of $\tau^*$. For simplicity, also suppose all $d_i$'s are nonzero. We define the *oracle estimator* as

$$(\widehat{\tau}^{or}, \widehat{\beta}^{or}) = \arg\min_{\tau,\beta}\{\|y - D\tau - X\beta\|_2^2 : \tau_i = 0 \text{ for } i \in S^c\}.$$

Then we have

$$\begin{cases} \widehat{\beta}^{or} = (X_{S^c}^\top X_{S^c})^{-1} X_{S^c}^T y_{S^c} \\ \widehat{\tau}_i^{or} = (y_i - x_i^\top \widehat{\beta}^{or})/d_i, \text{ for } i \in S \\ \widehat{\tau}_i^{or} = 0, \text{ for } i \in S^c \end{cases}$$

where $S^c$ is the complement of $S$ in $\{1, \ldots, n\}$, $X_{S^c}$ is the submatrix containing rows of $X$ indexed by $S^c$, and $y_{S^c}$ is the subvector containing components of $y$ indexed by $S^c$. Replacing $y = D\tau^* + X\beta^* + \varepsilon$ into the solutions above, we can rewrite

$$\widehat{\beta}^{or} = (X_{S^c}^\top X_{S^c})^{-1} X_{S^c}^T (X_{S^c}\beta^* + \varepsilon_{S^c}) = \beta^* + (X_{S^c}^\top X_{S^c})^{-1} X_{S^c}^T \varepsilon_{S^c},$$

and

$$\widehat{\tau}_i^{or} = \tau_i^* + \frac{x_i^\top(\beta^* - \widehat{\beta}^{or})}{d_i} + \frac{\varepsilon_i}{d_i}.$$

Note that $|S^c| \to \infty$ as $n \to \infty$. By the central limit theorem, we have

$$\sqrt{|S^c|}(\widehat{\beta}^{or} - \beta^*) \to N_p(\mathbf{0}, \sigma_\varepsilon^2 \Sigma_x^{-1}).$$

Further, if we make an assumption on the generation of heterogeneity such that the heterogeneity proportion $|S|/n$ converges to some value $\delta \in [0, 0.5)$, then $\sqrt{|S^c|}$ can be replaced by $\sqrt{n}$ in the above form with the asymptotic covariance discounted by a multiplier $1/(1-\delta)$. On the other hand, although the second term containing $\beta^* - \widehat{\beta}^{or}$ in the expression of $\widehat{\tau}_i^{or}$ vanishes eventually, the variability of its last term $\frac{\varepsilon_i}{d_i}$ remains unchanged no matter how large $n$ is!

The oracle estimator can be treated as a theoretical benchmark for our approach, so its properties give us some clues about the challenges we are facing. For HDR, therefore, trying to show consistency in estimating $\tau$ is simply unrealistic; moreover, the uncontrolled variability in $\widehat{\tau}$ may also affect the estimation of $\beta$. With this in mind, we consider important conditions on the penalty $\rho_\lambda$ and the loss $\mathcal{L}_n$, which give rise to the statistical guarantee to be discussed in the next section.

### 4.2.2 Nonconvex penalties

An important question for our method is what penalty functions should be used, especially the penalty $\rho_{\lambda_1}$ for $\tau_i$'s. The $L_1$ penalty $\rho_{\lambda_1}(t) = \lambda_1|t|$ applies uniform thresholding to all values of $\tau_i$'s so it tend to over-shrink large values, but data heterogeneity can often follow an irregular pattern with large deviation values. As a result, the $L_1$ penalty that leads to nonnegligible biases may not be able to accurately identify heterogeneous observations. As shown in our simulation experiments, $L_1$ penalty often falls into two extreme situations: it either discovers no heterogeneity at all or yields a large number of false discoveries. In comparison, nonconvex penalties such as SCAD (Fan and Li, 2001) and MCP (Zhang, 2010) impose no shrinkage on large values and usually enforce sparsity more aggressively, hence they are better suited for this problem.

The SCAD penalty takes the form

$$\rho_{\lambda,\gamma}(t) = \lambda \int_0^t \min\{1, (\gamma - x/\lambda)_+/(\gamma - 1)\}dx, \quad \gamma > 2,$$

and the MCP penalty is

$$\rho_{\lambda,\gamma}(t) = \lambda \int_0^t (1 - x/(\gamma\lambda))_+ dx, \quad \gamma > 1.$$

It is worth noting that both penalties have an extra parameter $\gamma$ that controls the concavity of the function. Actually, both penalties converge to $L_1$ when $\gamma \to \infty$. These three penalties will be compared thoroughly in the numerical experiments.

Our theory applies to more general conditions on the penalty function that encompass all three penalties mentioned here. We assume $\rho_\lambda$ satisfies the following:

**Assumption 1.**

(i) $\rho_\lambda$ is symmetric around 0 and $\rho_\lambda(0) = 0$.

(ii) For $t > 0$, $\rho_\lambda(t)$ is nondecreasing in $t$.

(iii) For $t > 0$, the function $t \mapsto \frac{\rho_\lambda(t)}{t}$ is nonincreasing in $t$.

(iv) $\rho_\lambda$ is differentiable for all $t \neq 0$, and subdifferentiable at $t = 0$ with the subgradients at 0 bounded by $\lambda$ in magnitude, i.e. $\partial\rho_\lambda(0) \subseteq [-\lambda, \lambda]$.

It is straightforward to verify that $L_1$, SCAD and MCP all satisfy the conditions of Assumption 1.

### 4.2.3 Restricted strong convexity

In the most general form of HDR, we only require the loss $\mathcal{L}_n$ to be differentiable but does not require it to be convex. Even if $\mathcal{L}_n$ is indeed convex, it can never be strongly convex in the high-dimensional setting of our problem. Instead, we consider a weaker condition on $\mathcal{L}_n$ known as *restricted strong convexity* (RSC):

$$\mathcal{E}_n(\Delta) = \langle \nabla\mathcal{L}_n(\theta^* + \Delta) - \nabla\mathcal{L}_n(\theta^*), \Delta\rangle \geq \alpha\|\Delta_2\|_2^2 - \kappa\frac{\log p}{n}\|\Delta\|_1^2, \tag{4.4}$$

where $\theta^* = (\tau^{*\top}, \beta^{*\top})^\top$ denotes the true value of $\theta$, $\Delta = (\Delta_1^\top, \Delta_2^\top)^\top$ with $\Delta_1 \in \mathbb{R}^n$, $\Delta_2 \in \mathbb{R}^p$, $\alpha > 0$ and $\kappa \geq 0$.

How do we understand this condition? Note that for $\delta \in \mathbb{R}^{n+p}$, by the mean value theorem we have

$$\mathcal{L}_n(\theta^* + \delta) - \mathcal{L}_n(\theta^*) = \langle \nabla \mathcal{L}_n(\theta^* + c\delta), \delta \rangle ,$$

for some $c \in [0, 1]$. By rewriting $\Delta = c\delta$, this implies

$$\mathcal{L}_n(\theta^* + \Delta/c) = \mathcal{L}_n(\theta^*) + \langle \nabla \mathcal{L}_n(\theta^*), \Delta \rangle / c + \langle \nabla \mathcal{L}_n(\theta^* + \Delta) - \nabla \mathcal{L}_n(\theta^*), \Delta \rangle / c.$$

That is, the RSC condition (4.4) essentially imposes a lower bound on the remainder term of the first-order Taylor expansion of $\mathcal{L}_n$ at $\theta^*$. Hence when $\mathcal{L}_n$ is convex, this remainder is always nonnegative, then (4.4) holds trivially for $\frac{\|\Delta_2\|_2}{\|\Delta\|_1} \leq \sqrt{\frac{\kappa \log p}{\alpha n}}$, and the RSC condition is only enforced on a cone set $\left\{ \Delta : \frac{\|\Delta_2\|_2}{\|\Delta\|_1} > \sqrt{\frac{\kappa \log p}{\alpha n}} \right\}$.

Similar conditions have been discussed in previous literature such as Agarwal et al. (2012), Loh and Wainwright (2015). Unlike the past work, the first term of our RSC condition (4.4) only involves $\Delta_2$, the part of $\Delta$ corresponding to $\beta$, instead of the entire $\Delta$. This is due to the special structure of our problem. In our formulation, $X$ is dense while $D$ is diagonal that contains no extra information beyond $X$, hence it seems only reasonable to assume RSC on the $\beta$ part corresponding to $X$. Besides, it would be hardly possible to establish the probabilistic guarantee as stated in the next section by using $\Delta$ in the first term of the right side.

### 4.3 Statistical properties

With the above setup, the nonconvex optimization problem of (4.2) may have multiple local optima. Nonconvex optimization is computationally intractable in general, and existing iterative algorithms such as coordinate descent and gradient descent are only guaranteed to convergence to a local optimizer. Hence instead

of focusing on the global optimizer, it is probably more interesting to ask what statistical properties a local one can have. In this section, we establish our main statistical guarantee for generic loss functions and a probabilistic corollary for the least squares loss. These results apply to all local optimizers of (4.2).

In addition, it is worth noting that the results cover both types of specification for $(d_i, \tau_i)$ as described in the introduction. For example, suppose the underlying source of heterogeneity is indeed a covariate $X_j$, i.e $d_i = x_{ij}$ for all $i$. Since we may not have any prior knowledge about that, we could simply choose the "misspecified" model with $d_i' = 1$, for which the corresponding deviation effect is $\tau_i' = x_{ij}\tau_i$. Our results cover both the model with correctly specified $d_i = x_{ij}$ and the model with "misspecified" $d_i' = 1$.

### 4.3.1 Main result

As clear in the previous discussion, even in the imaginary scenario where we know exactly which observations have nonzero $\tau_i$, the $\widehat{\tau}^{or}$ of the oracle estimator is not consistent for $\tau$. Thus a major theoretical challenge is to characterize the statistical precision of the estimated $\beta$ in spite of the impreciseness of the estimated $\tau$. Our main theorem is deterministic in nature, providing an error bound for the $\widetilde{\beta}$ component of any local optimizer $\widetilde{\theta}$ of problem (4.2), which guarantees $\widetilde{\beta}$ lies close to the underlying truth $\beta^*$ under mild conditions.

**Lemma 4.1.** *Suppose $\widetilde{\theta}$ is a local minimizer of* (4.2)*. Then there exists $\widetilde{w} \in \partial\rho_\lambda(\widetilde{\theta})$ such that*

$$\langle \nabla \mathcal{L}_n(\widetilde{\theta}) + \widetilde{w}, \theta - \widetilde{\theta} \rangle \geq 0, \quad \textit{for all feasible } \theta \in \mathbb{R}^{n+p}. \tag{4.5}$$

*When $\widetilde{\theta}$ is an interior point of the feasible region, this result reduces to the usual zero subgradient condition*

$$\mathbf{0} \in \nabla \mathcal{L}_n(\widetilde{\theta}) + \partial\rho_\lambda(\widetilde{\theta}).$$

*Proof.* Let $g(\theta) = \|\theta\|_1 - R$, then $g$ is convex and (4.2) clearly satisfies the Slater condition. By Theorem 2.2, the Lagrangian can be simplified as

$$L(\theta, r) = \mathcal{L}_n(\theta) + \rho_\lambda(\theta) + rg(\theta),$$

where $r \geq 0$. Then Theorem 2.1 implies that there exists $\widetilde{w} \in \partial\rho_\lambda(\widetilde{\theta})$, $\widetilde{v} \in \partial g(\widetilde{\theta})$ and Lagrangian multiplier $\widetilde{r} \geq 0$ such that

$$\widetilde{r}g(\widetilde{\theta}) = 0, \tag{4.6}$$

and

$$\mathbf{0} = \nabla\mathcal{L}_n(\theta) + \widetilde{w} + \widetilde{r}\widetilde{v}. \tag{4.7}$$

Let $\theta$ be any feasible point for (4.2). By convexity of $g$, we have

$$g(\theta) - g(\widetilde{\theta}) \geq \langle \widetilde{v}, \theta - \widetilde{\theta} \rangle. \tag{4.8}$$

Then combining (4.6), (4.7) and (4.8) yields

$$\begin{aligned} \langle \nabla\mathcal{L}_n(\widetilde{\theta}) + \widetilde{w}, \theta - \widetilde{\theta} \rangle &= -\widetilde{r}\langle \widetilde{v}, \theta - \widetilde{\theta} \rangle \\ &\geq -\widetilde{r}(g(\theta) - g(\widetilde{\theta})) \\ &= -\widetilde{r}g(\theta) \\ &\geq 0. \end{aligned}$$

□

This lemma allows us to relate a local minimizer $\widetilde{\theta}$ with any other feasible point $\theta$ through an inequality. In particular, $\theta$ here can be replaced by the underlying truth $\theta^*$. Based on this result and other key conditions on the penalty and the loss, we derive the following main result

**Theorem 4.2.** *Assume the penalty $\rho_\lambda$ satisfies Assumption 1, the loss $\mathcal{L}_n$ satisfies the RSC condition* (4.4)*, and $\theta^*$ is feasible for problem* (4.2)*. For any given* $0 <$

$\epsilon < 1$, *suppose* $n, p, R$ *together satisfy*

$$\cdot \frac{R^2 \log(n+p)}{n} \leq \frac{\min(\alpha^2, 1)}{16 \max(\kappa^2, 1)} \epsilon^4, \tag{4.9}$$

*and consider any* $\lambda$ *such that*

$$\|\nabla \mathcal{L}_n(\theta^*)\|_\infty \leq \lambda \leq \frac{\alpha \epsilon^2}{8R}. \tag{4.10}$$

*Then for any local minimizer* $\widetilde{\theta} = (\widetilde{\tau}^\top, \widetilde{\beta}^\top)^\top$ *of* (4.2)*, its* $\widetilde{\beta}$ *component has an error bound*

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \epsilon.$$

**Remark.** Theorem 4.2 bounds $\widetilde{\beta}$ with essentially $\|\widetilde{\beta} - \beta^*\|_2 = O((\frac{R^2 \log(n+p)}{n})^{1/4})$, which does not involve $k = \|\beta^*\|_0$ and differs from the scale $O(\sqrt{\frac{k \log p}{n}})$ in Loh and Wainwright (2015) for standard nonconvex penalized regression. This can be considered as a compromise for the consideration of heterogeneous population for which the standard (penalized) estimates will be always biased.

Beside, we emphasize again Theorem 4.2 is entirely deterministic. It is intended to encompass various choices of loss functions such as least squares loss, Huber loss and Tukey's bisquare loss. For least squares loss in particular, a probabilistic result will be derived subsequently, where we establish the required conditions with high probability, including notably the RSC condition.

*Proof.* For brevity, denote $\widetilde{\Delta}_1 = \widetilde{\tau} - \tau^*$, $\widetilde{\Delta}_2 = \widetilde{\beta} - \beta^*$ and $\widetilde{\Delta} = (\widetilde{\Delta}_1^\top, \widetilde{\Delta}_2^\top)^\top$. We prove the error bound by contradiction. Assume $\|\widetilde{\Delta}_2\|_2 > \epsilon$. The feasibility of $\theta^*$, $\widetilde{\theta}$ and the triangle inequality imply that

$$\|\widetilde{\Delta}\|_1 \leq \|\widetilde{\theta}\|_1 + \|\theta^*\|_1 \leq 2R. \tag{4.11}$$

And with $\|\widetilde{\Delta}_2\|_2 > \epsilon$, it follows from the RSC (4.4) with $\theta = \widetilde{\theta}$ and the assumed

lower bound on $n$ (4.9) that

$$\begin{aligned}\langle \nabla \mathcal{L}_n(\widetilde{\theta}) - \nabla \mathcal{L}_n(\theta^*), \widetilde{\Delta} \rangle &\geq \alpha\epsilon \|\widetilde{\Delta}_2\|_2 - \kappa \frac{\log(n+p)}{n} \cdot 2R \|\widetilde{\Delta}\|_1 \\ &\geq \alpha\epsilon \|\widetilde{\Delta}_2\|_2 - \frac{\epsilon^2}{2}\sqrt{\frac{\log(n+p)}{n}} \|\widetilde{\Delta}\|_1. \end{aligned} \tag{4.12}$$

Let $\widetilde{w} \in \partial \rho_\lambda(\widetilde{\theta})$ be the subgradient satisfying the inequality (4.5) in Lemma 4.1. Combining this with the fact that $\theta^*$ is feasible and the inequality (4.12) with $\theta = \theta^*$ yields

$$\langle -\widetilde{w} - \nabla \mathcal{L}_n(\theta^*), \widetilde{\Delta} \rangle \geq \alpha\epsilon \|\widetilde{\Delta}_2\|_2 - \frac{\epsilon^2}{2}\sqrt{\frac{\log(n+p)}{n}} \|\widetilde{\Delta}\|_1. \tag{4.13}$$

By Hölder's inequality and triangle inequality, we also have

$$\begin{aligned}\langle -\widetilde{w} - \nabla \mathcal{L}_n(\theta^*), \widetilde{\Delta} \rangle &\leq \left(\|\widetilde{w}\|_\infty + \|\nabla \mathcal{L}_n(\theta^*)\|_\infty\right) \|\widetilde{\Delta}\|_1 \\ &\leq 2\lambda \|\widetilde{\Delta}\|_1 \end{aligned} \tag{4.14}$$

Combining (4.13) and (4.14) and rearranging terms then yields the following

$$\begin{aligned}\|\widetilde{\Delta}_2\|_2 &\leq \frac{\|\widetilde{\Delta}\|_1}{\alpha\epsilon}\left(2\lambda + \frac{\epsilon^2}{2}\sqrt{\frac{\log(n+p)}{n}}\right) \\ &\leq \frac{2R}{\alpha\epsilon}\left(2\lambda + \frac{\epsilon^2}{2}\sqrt{\frac{\log(n+p)}{n}}\right) \\ &= \frac{4R\lambda}{\alpha\epsilon} + \frac{R\epsilon}{\alpha}\sqrt{\frac{\log(n+p)}{n}}. \end{aligned} \tag{4.15}$$

Then by our choice of $\lambda$ in (4.10) and the lower bound on $n$ in (4.9), we have respectively

$$\frac{4R\lambda}{\alpha\epsilon} \leq \frac{\epsilon}{2}, \quad \frac{R\epsilon}{\alpha}\sqrt{\frac{\log(n+p)}{n}} \leq \frac{\epsilon^3}{2} \leq \frac{\epsilon}{2} \tag{4.16}$$

Combining (4.15) and (4.16) yields the $\ell_2$ bound on $\widetilde{\beta}$:

$$\|\widetilde{\beta} - \beta^*\|_2 = \|\widetilde{\Delta}_2\|_2 \leq \epsilon.$$

□

### 4.3.2 Least squares loss

Before moving on to the corollary, we need to first introduce the notion of sub-Gaussian random variables and matrices (Vershynin, 2010).

**Definition 4.1.** We say a random variable $X$ is *sub-Gaussian with parameter* $K$ if

$$K = \sup_{p \geq 1} p^{-1/2} (E|X|^p)^{1/p} < \infty.$$

$K$ is also called the *sub-Gaussian norm* of $X$, denoted as $\|X\|_{\psi_2}$.

Thus given a probability space, the class of sub-gaussian random variables forms a normed space. In addition, a sub-Gaussian $X$ has the following *tail decay property*: there exists $c > 0$ such that

$$P(|X| > t) \leq \exp(1 - ct^2/\|X\|_{\psi_2}^2), \quad \forall t \geq 0.$$

**Definition 4.2.** We say a random matrix $M \in R^{m \times n}$ is sub-Gaussian with parameter $(\Sigma, \sigma)$ if

(i) each row $m_i \in \mathbb{R}^n$ is sampled independently with covariance $\Sigma$, and

(ii) for any unit vector $u \in \mathbb{R}^n$, the random variable $u^\top m_i$ is sub-Gaussian with parameter $\sigma$.

Now consider the least squares loss $\ell(t) = t^2/2$, thus the entire loss part is $\mathcal{L}(\theta) = \mathcal{L}(\tau, \beta) = \frac{1}{2n}\|y - D\tau - X\beta\|_2^2$. Based on Theorem 4.2 and two auxiliary lemmas in Appendix D, we derive a probabilistic corollary for this case:

**Corollary 4.3.** *Let* $\mathcal{L}(\theta) = \frac{1}{2n}\|y - D\tau - X\beta\|_2^2$, *where* $D = diag(d)$. *Suppose* $d$ *is sub-Gaussian with parameter* $\sigma_d$, $X$ *is sub-Gaussian with parameter* $(\Sigma_x, \sigma_x)$, *and* $\varepsilon_i$*'s are i.i.d sub-Gaussian with parameter* $\sigma_\varepsilon$. *Suppose* $R$ *is chosen such that* $\theta^*$ *is feasible, and* $\lambda = c\sqrt{\log(n+p)/n}$ *for some* $c > 0$. *Assume* $n \gtrsim \log(n+p)$, *then the* $\widetilde{\beta}$ *component of any local optimum* $\widetilde{\theta}$ *satisfies the error bound*

$$\|\widetilde{\beta} - \beta^*\|_2 \leq C \left( \frac{R^2 \log(n+p)}{n} \right)^{1/4}$$

*for some constant* $C > 0$ *with probability at least* $1 - c_1 \exp(-c_2 \log(n+p))$.

**Remark.** In most applications, we expect $p$ to be growing slower than $e^n$ so commonly we have $\log(n+p) = o(n)$. Hence the bound shrinks to zero as $n \to \infty$ as long as $R = o(\sqrt{n/\log(n+p)})$, and then the corollary implies any local minimum $\widetilde{\beta}$ of (4.2) is consistent in estimating $\beta^*$.

*Proof.* In order to apply Theorem 1, it suffices to verify the RSC condition and the validity of the choice of $\lambda$ with high probability. Denote $G = \frac{1}{n} X^\top X$. Then the left hand side of the RSC is

$$\begin{aligned}
\mathcal{E}_n(\Delta) &= \frac{1}{n}\Delta^\top (D\,X)^\top (D\,X)\Delta \\
&= \frac{1}{n}\Delta_1^\top D^2 \Delta_1 + \frac{2}{n}\Delta_1^\top DX\Delta_2 + \Delta_2^\top G\Delta_2 \\
&\geq \frac{2}{n}\Delta_1^\top DX\Delta_2 + \Delta_2^\top G\Delta_2 \\
&\geq \frac{2}{n}\Delta_1^\top DX\Delta_2 + \Delta_2^\top \Sigma_x\Delta_2 - |\Delta_2^\top (G-\Sigma_x)\Delta_2| \\
&\geq -|\frac{2}{n}\Delta_1^\top DX\Delta_2| + \lambda_{\min}(\Sigma_x)\|\Delta_2\|_2^2 - |\Delta_2^\top (G-\Sigma_x)\Delta_2|
\end{aligned}$$

Lemma D.1 implies that with probability $1 - c_1 \exp(-c_2 \log(n+p))$,

$$\begin{aligned}
|\frac{2}{n}\Delta_1^\top DX\Delta_2| &\leq \frac{c\log(n+p)}{n}\|\Delta_1\|_1\|\Delta_2\|_2 \\
&\leq \frac{c\log(n+p)}{n}\|\Delta_1\|_1\|\Delta_2\|_1 \\
&\leq \frac{c\log(n+p)}{n}\|\Delta\|_1^2
\end{aligned}$$

And by Lemma 12 of Loh and Wainwright (2012) with $s = \frac{n}{\log p}$, the last term $|\Delta_2^\top (G-\Sigma_x)\Delta_2|$ is bounded by

$$\frac{\lambda_{\min}(\Sigma_x)}{2}\|\Delta_2\|_2^2 + \frac{c'\log p}{n}\|\Delta_2\|_1^2$$

with probability at least $1 - c_3\exp(-c_4 n)$. Combining these results we have

$$\mathcal{E}_n(\Delta) \geq \frac{\lambda_{min}(\Sigma_x)}{2}\|\Delta_2\|_2^2 - (c+c')\frac{\log(n+p)}{n}\|\Delta\|_1^2.$$

It remains to verify the validity of the specification of $\lambda$. By Lemma D.2 we have

$$\|\nabla \mathcal{L}_n(\theta^*)\|_\infty \leq c'' \sqrt{\frac{\log(n+p)}{n}},$$

with probability at least $1 - c_5 \exp(-c_6 \log(n+p))$. Hence the choice of $\lambda$ is valid. Consequently, the error bound follows by applying Theorem 1.

□

### 4.4 Computation

Denote $Z = (D\, X)$, and $\theta = (\tau, \beta)$. Then (4.2) can be rewritten as

$$\min_{\|\theta\|_1 \leq R} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i - z_i^\top \theta)^2 + \rho_\lambda(\theta).$$

So far the constraint $\|\theta\|_1 \leq R$ has been used mainly as a theoretical technique. Suppose we relax the constraint by using a bound $R$ large enough to contain all the local minima, and consider only the least squares loss $\ell(t) = t^2/2$ and MCP, SCAD or $L_1$ penalty. Then the solutions can be computed via a publicly available R package `ncvreg` that implements a coordinate descent algorithm (Breheny and Huang, 2011). By default `ncvreg` internally standardizes the input data. Since $D$ is diagonal, we prefer to use the raw data as is instead of standardizing it, computed through the `ncvreg_raw` function from the package. If it is desirable to standardize the covariates in $X$, we simply perform the standardization beforehand, then augment the data with $D$ and use $Z = (D\, X)$ as the input data matrix.

### 4.5 Simulation studies

In this section, we investigate the empirical performances of the proposed methods through simulation experiments.

In our model we need to estimate $\tau$ and $\beta$ which has a total dimension of $n+p$. In this high-dimensional setting we adopt the modified Bayesian Information

Criterion (Wang et al., 2007) and select the tuning parameter $\lambda$ by minimizing

$$\mathrm{BIC}(\lambda) = \log\left[\|y - D\widehat{\tau} - X\widehat{\beta}\|_2^2/n\right] + C_{n,p}\frac{\log n}{n}(\|\widehat{\tau}\|_0 + \|\widehat{\beta}\|_0),$$

where $C_{n,p}$ is a positive number that depends on $n$ and $p$. When $C_{n,p} = 1$, the modified BIC reduces to the traditional BIC (Schwarz, 1978). In this paper, we use $C_{n,p} = c\log(\log(n+p))$ as suggested in Wang et al. (2009) when the number of predictors diverges with the sample size. We set $c = 0.5$ and use a fixed value for $\gamma$ for MCP and SCAD penalties in the experiments.

In our analysis, we compare the performances of the following methods: the proposed heterogeneity discovery regression (HDR) with three different penalties, MCP, SCAD and $L_1$, denoted as H-MCP, H-SCAD, H-$L_1$ respectively; three baseline regression methods, least squares (LS) and two robust alternatives, Huber loss regression (Huber) and least absolute deviations regression (LAD); in the case that the source of heterogeneity is some covariate ($d_i = x_{ij}$), we will also include an extra "misspecified" model that pretends $d_i = 1$ and imposes MCP penalties on $\tau_i$'s, which we denote as H-MIS. Moreover, when the covariates are high-dimensional, for the baseline methods we also impose the $L_1$ penalty on the coefficients and similarly select the tuning parameter $\lambda$ using the modified BIC. The HDR methods were fitted by R package `ncvreg` and the baseline methods were fitted by R package `hqreg`, both available on CRAN. In every experiment, we use `ncvreg` the default penalty parameter $\gamma = 3$ for MCP and $\gamma = 3.7$ for SCAD in the HDRs, and set the threshold parameter to $\mathrm{IQR}(y)/10$ for Huber loss regression.

Example 1

We consider the simple regression setting where $p = 2$ with $X = (1\, x)$ and provide visualization for the fitted models.

**Case 1.** We simulate the data from the model

$$y_i = \tau_i + \beta_0 + x_i\beta_1 + \varepsilon_i, \quad i = 1, \ldots, n.$$

where $x_i$'s are i.i.d. sampled from Uniform$[0, 1]$ and the random errors $\varepsilon_i$'s are from $\mathrm{N}(0, 0.5^2)$. We set $n = 300, \beta_0 = 1, \beta_1 = 5$, and generate the deviation terms $\tau_i$ by setting $\tau_i = 10 \cdot I(x_i < 0.2)$ so that the outliers are all on one side.

Figure 4.1 shows the lines $y = \widehat{\beta}_0 + x\widehat{\beta}_1$ fitted by different methods. We observe that the overlapped fits by H-MCP and H-SCAD best match the overall trend of the data. In comparison, the fitted lines by Huber, LAD and H-$\mathrm{L}_1$ also show some extent of robustness towards outliers yet still slightly affected, while the fitted line by LS completely deviates from the overall trend due to high sensitivity of the outliers.

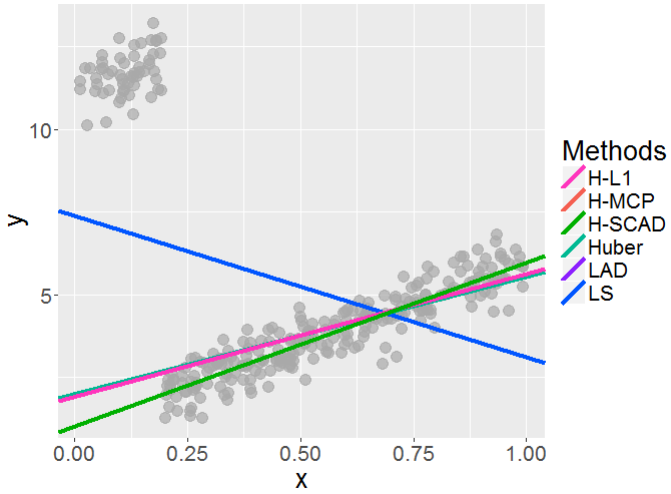


Figure 4.1: Fitted lines on Example 1 – Case 1

More precisely, we use the $\mathrm{L}_2$ error $\|\widehat{\beta} - \beta\|_2$ to measure estimation performances on $\beta$, as reported in Table 4.1. H-MCP and H-SCAD have much smaller estimation errors compared to the other methods, consistent with what we find in Figure 4.1. Among the HDRs, although our theoretical error bounds apply to all

three penalties MCP, SCAD and $L_1$, the two nonconvex penalties MCP and SCAD outperform the $L_1$ penalty numerically by quite a large margin.

| Method | $\|\widehat{\beta} - \beta\|_2$ |
|---|---|
| H-MCP | 0.047 |
| H-SCAD | 0.051 |
| H-$L_1$ | 1.828 |
| Huber | 1.786 |
| LAD | 1.762 |
| LS | 11.279 |

Table 4.1: Estimation of $\beta$ for Example 1 – Case 1

| Method | FDP | MP |
|---|---|---|
| H-MCP | 0 | 0 |
| H-SCAD | 0 | 0 |
| H-$L_1$ | 0.715 | 0 |

Table 4.2: Heterogeneity discovery performances for Example 1 – Case 1

Further we investigate the heterogeneity discovery performances which are not shown in the plot. We report the $L_2$ error $\|\widehat{\tau} - \tau\|_2$, False Discovery Proportion (FDP), Miss Proportion (MP) of the six methods in Table 4.2. FDP and MP are computed using the formula:

$$\text{FDP} = \frac{\text{FP}}{\text{TP+FP}}, \qquad \text{MP} = \frac{\text{FN}}{\text{TP+FN}},$$

where a true positive (TP) is a correctly identified observation with nonzero $\tau_i$, a true negative (TN) is a correctly identified one with zero $\tau_i$, a false positive (FP) is

an observation with the truth $\tau_i = 0$ but the estimate $\widehat{\tau}_i \neq 0$, and a false negative (FN) is the opposite with $\tau_i \neq 0$ but $\widehat{\tau}_i = 0$. We observe that H-MCP and H-SCAD have both zero FDP and MP, showing that these two methods perfectly distinguish between outliers and normal cases in the settings of this example. On the other hand, H-$L_1$ has a large FDP, showing that it has failed to exclude the noises from the true signals.

**Case 2.** In this case, we consider heterogeneity on the slope $\beta_1$ instead, i.e. $D = \text{diag}(x_1, \ldots, x_n)$. We simulate the data from

$$y_i = \beta_0 + x_i(\beta_1 + \tau_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where we change the setting by generating $x_i$ from $\text{N}(0, 1)$ and $\tau_i$ randomly with $P(\tau_i = 0) = 0.8$ and $P(\tau_i = 50) = 0.2$.

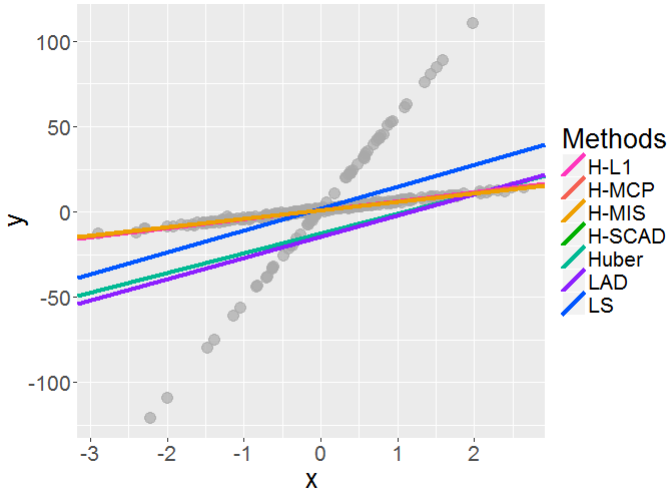


Figure 4.2: Fitted lines by different methods for Example 1 – Case 2

Figure 4.2 shows the fitted lines in this case. We find that all the HDRs fit the data pretty well and generate almost overlapped models and LS is again heavily affected by the outliers as expected. But surprisingly, Huber and LAD appear to perform even worse than LS. Table 4.3 provides further support to our impression

from Figure 4.2. We observe that the the estimation errors of the HDRs are again much smaller than those of the baseline methods, and the errors of Huber, LAD are indeed greater than that of LS – these two supposedly robust alternatives to the least squares somehow become even less robust in this case. It is also noteworthy that H-MIS actually has the smallest estimation error, despite the fact that H-MIS uses $D = I_n$ which is a "misspecification" of the source of heterogeneity.

Table 4.4 reports the heterogeneity discovery performances in this case. Perhaps affected by the randomness in the $D$ matrix, the FDP and MP values for the previously perfect H-MCP and H-SCAD are inflated in this case. In comparison, H-MIS has much smaller values for both statistics. It seems we are better off using $D = I_n$, even when the heterogeneity actually comes from a particular nontrivial random variable $x$.

| Method | $\|\widehat{\beta} - \beta\|_2$ |
|---|---|
| H-MIS | 0.041 |
| H-MCP | 0.160 |
| H-SCAD | 0.161 |
| H-$L_1$ | 0.354 |
| Huber | 19.528 |
| LAD | 17.506 |
| LS | 7.909 |

Table 4.3: Estimation of $\beta$ for Example 1 – Case 2

Example 2

In this example we consider three different scenarios of heterogeneity in the setting of low-dimensional multivariate regression.

| Method | FDP | MP |
|---|---|---|
| H-MIS | 0.033 | 0.017 |
| H-MCP | 0.25 | 0.15 |
| H-SCAD | 0.239 | 0.15 |
| H-$L_1$ | 0.354 | 0.15 |

Table 4.4: Heterogeneity discovery performances for Example 1 – Case 2

**Case 1.** We simulate data from the following model:

$$y_i = \tau_i + \beta_0 + \sum_{j=1}^{5} x_{ij}\beta_j + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $x_i = (x_{i1}, \ldots, x_{i5})^\top$ are generated from a multivariate normal distribution with mean 0, variance 1 and pairwise correlation 0.25, and the error terms are from $\mathrm{N}(0, 0.1^2)$. For $\beta = (\beta_0, \ldots, \beta_5)$, we set $\beta_0 = 1$ and generate $\beta_1, \ldots, \beta_5$ independently from Uniform$(0.5, 1)$. And again we set $n = 300$, and randomly generate $\tau_i$'s with the percentage of heterogeneity fixed at 0.3, i.e. $P(\tau_i = 0) = 0.7$ and $P(\tau_i = \Gamma) = 0.3$, where $\Gamma$ is the deviation amount that we allow to vary from 0 to 3. We consider the average performances over 100 repetitions of the simulation settings.

Figure 4.3 shows the trend of $L_2$ estimation errors for $\beta$ as the deviation $\Gamma$ ranges from 0 to 3. While the errors by different methods look similar when deviation is close to zero, they display very different traits as this amount increases. The performances of the methods can be classified into four tiers: H-MCP and H-SCAD constitutes the top tier which has consistently small estimation error, H-$L_1$ alone is the second tier, Huber and Quantier are the third tier, and LS is the worst. While the estimation error for LS is almost linearly increasing in the deviation $\Gamma$, all the other methods seem to stabilize when the deviation passes 1, 100% relative to $\beta_0$.

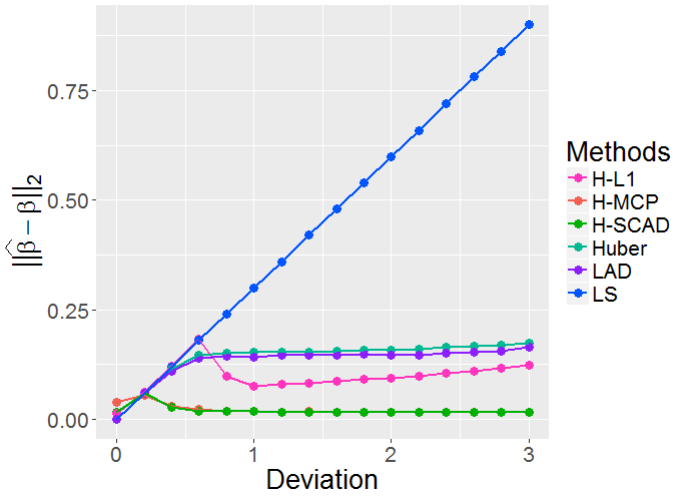


Figure 4.3: Estimation error on $\beta$ for Example 2 – Case 1 for varying deviation amounts, averaged over 100 repetitions

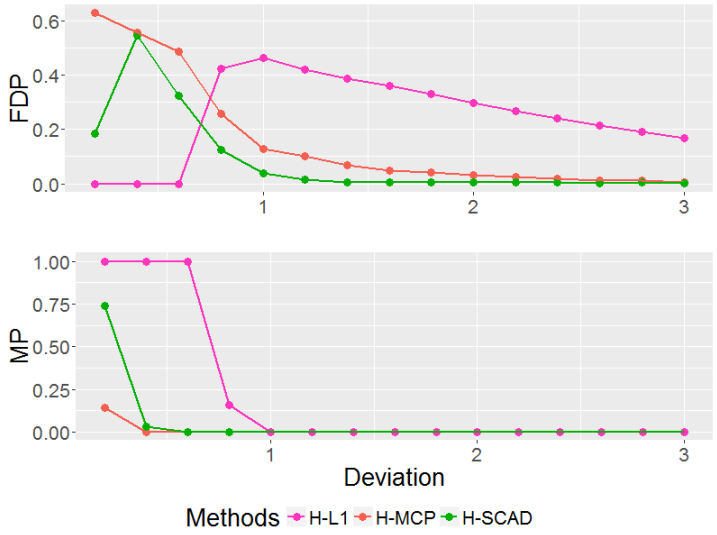


Figure 4.4: Heterogeneity discovery for Example 2 – Case 1 for varying deviation amounts, averaged over 100 repetitions

Next we compare the three HDRs based on their heterogeneity discovery performances, as shown in Figure 4.4. When deviation is close to zero, none of the three behave satisfactorily. They have either a small miss percentage with a large amount of false discoveries or the opposite. When the deviation passes 1, all three have zero MP, and their FDPs steadily decline. For this aspect, H-SCAD seem to slightly outperform H-MCP in general and H-$L_1$ is clearly trailing.

**Case 2.** We change the setting of Case 1 only in the generation of $\tau_i$'s. This time, we vary the percentage of heterogeneity $P(\tau_i \neq 0)$ from 0 to 0.45. In addition, we allow the $\tau_i$'s to take several values -1, 0.5, 2 with proportions 30%, 20% and 50% within the set of nonzero ones.

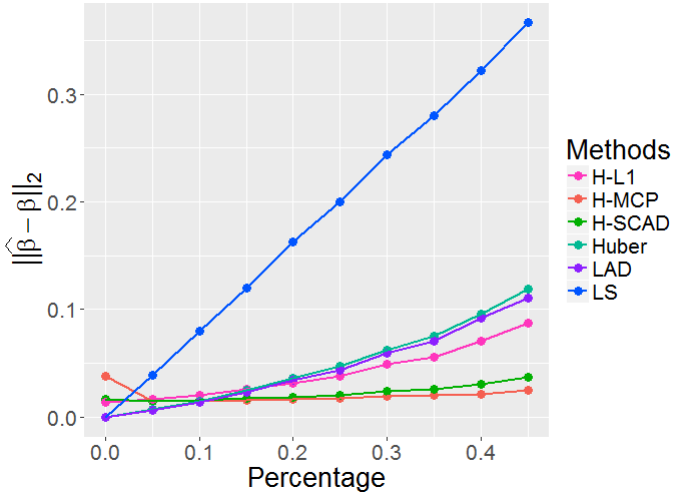


Figure 4.5: Estimation error on $\beta$ for Example 2 – Case 2 for varying percentage of heterogeneity, averaged over 100 repetitions

As revealed in Figure 4.5, in this case the estimation errors on $\beta$ by all the methods increase with the heterogeneity percentage, although at very different rates. The performance tiers are consistent with what we observe in Case 1. Interestingly, the baseline methods yield slighly more accurate estimation when the percentage is close to zero, but this tiny advantage disappears as the percentage increases and the performances of the HDRs, especially of H-MCP and H-SCAD, become dominant

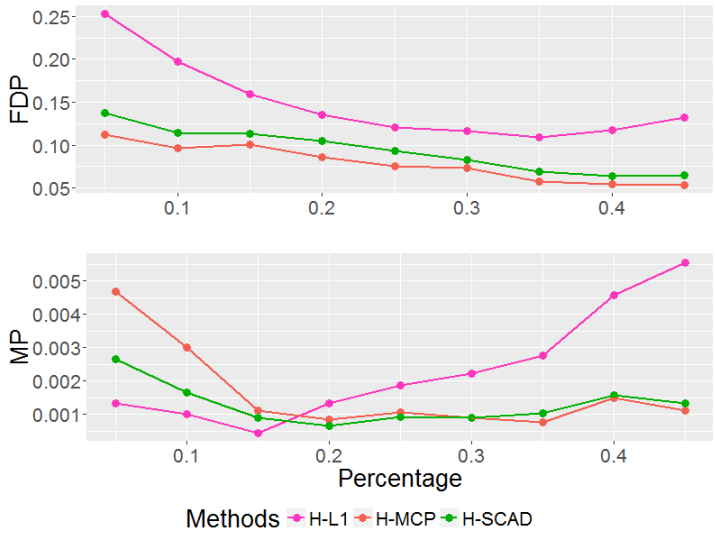


Figure 4.6: Heterogeneity discovery for Example 2 – Case 2 for varying percentage of heterogeneity, averaged over 100 repetitions

eventually. Figure 4.6 shows the heterogeneity discovery performances in this case, which tells a very different story from Figure 4.4 for the previous case. FDPs and MPs of the HDRs are drastically sensitive to the deviation amount in Case 1 but in this case they only change mildly as the percentage varies.

Example 3

In this example we introduce a categorical feature $u_i$ with levels "A", "B" into the multivariate regression setting of Example 2 and consider heterogeneity on the coefficients of $u_i$. We simulate data from this model:

$$y_i = \tau_i + I(u_i = \text{“A”}) \cdot \beta_A + I(u_i = \text{“B”}) \cdot \beta_B + \sum_{j=1}^{5} x_{ij}\beta_j + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $x_i$, $\varepsilon_i$ and $\beta_1, \ldots, \beta_5$ are generated using the same settings of example 2, but $\tau_i$ are generated with dependence on $u_i$: given $u_i$ = "A", $\tau_i$ can take values of 0 or 1 with $P(\tau_i = 1 | u_i = \text{“A”}) = 0.3$; given $u_i$ = "B", $\tau_i$ can take values of 0 or -1 with $P(\tau_i = -1 | u_i = \text{“B”}) = 0.2$. That is, the two levels "A", "B" have different distributions of heterogeneity. Besides, we generate $u_i$ from the two levels "A", "B"

with equal probabilities and set $\beta_A = 0$, $\beta_B = 0.3$.

Table 4.5 presents the performances of estimating $\beta$ based on 100 realizations of the simulation settings. We observe that H-MCP and H-SCAD again dominate the other methods, similar to what we have seen in Example 1. In particular, they both achieve almost unbiased estimation of $\beta_A$ and $\beta_B$. H-$L_1$ comes in second with inflated biases but still it allows us to easily differentiate between $\widehat{\beta}_A$ and $\widehat{\beta}_B$. The two robust regression methods Huber and LAD are clearly worse than these HDRs and their estimates $\widehat{\beta}_A$, $\widehat{\beta}_B$ become very close and indistinguishable. Finally, LS has terrible performances, which is completely fooled by the misleading outliers and actually estimates $\beta_A$ to be larger than $\beta_B$.

| Method | $\|\widehat{\beta} - \beta\|_2$ | $\widehat{\beta}_A$ | $\widehat{\beta}_B$ |
|---|---|---|---|
| H-MCP | 0.022 | 4e-5 | 0.300 |
| | (0.008) | (0.010) | (0.010) |
| H-SCAD | 0.022 | 0.002 | 0.300 |
| | (0.007) | (0.010) | (0.010) |
| H-$L_1$ | 0.139 | 0.073 | 0.258 |
| | (0.019) | (0.012) | (0.011) |
| Huber | 0.283 | 0.150 | 0.214 |
| | (0.035) | (0.019) | (0.014) |
| LAD | 0.274 | 0.146 | 0.216 |
| | (0.048) | (0.025) | (0.018) |
| LS | 0.579 | 0.297 | 0.102 |
| | (0.015) | (0.009) | (0.010) |

Table 4.5: Estimation of $\beta$ for Example 3. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

Table 4.6 reports the heterogeneity discovery performances of the three HDR

| Method | FDP | MP |
|---|---|---|
| H-MCP | 0.073 | 0 |
| | (0.120) | (0) |
| H-SCAD | 0.044 | 0 |
| | (0.037) | (0) |
| H-$L_1$ | 0.425 | 0 |
| | (0.074) | (0) |

Table 4.6: Heterogeneity discovery performances for Example 3. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

methods. While all three methods successfully discovery all the true positives, H-MCP and H-SCAD have much smaller FDPs than H-$L_1$, indicating they are much better at avoiding false positives. We also notice that H-SCAD has the smallest standard error, which means it is the most stable in this example. Figure 4.7 shows the solution paths of these three methods for $\tau_i$'s on one of the 100 realizations and illustrates why H-MCP and H-SCAD are performing better than H-$L_1$. In Figure 4.7, we find that the nonzero $\widehat{\tau}_i$'s of H-MCP and H-SCAD quickly converge and stabilize around 1 and -1 and it is easy to see that at a $\lambda$ value where $\log(\lambda) \approx -7$ we could perfectly separate the stable estimates of the truly nonzero $\tau_i$ from the noises arising from there. On the other hand, since the $L_1$ penalty imposes heavier shrinkage effect than MCP and SCAD, the nonzero $\widehat{\tau}_i$'s of H-$L_1$ slowly increase their magnitudes within the range of $\lambda$ rather than quickly stabilize at some levels. Besides, the noises start to appear earlier than what we discover in the paths of H-MCP and H-SCAD. Hence it is impossible to obtain a $\lambda$ value that separates the true signals from noises.

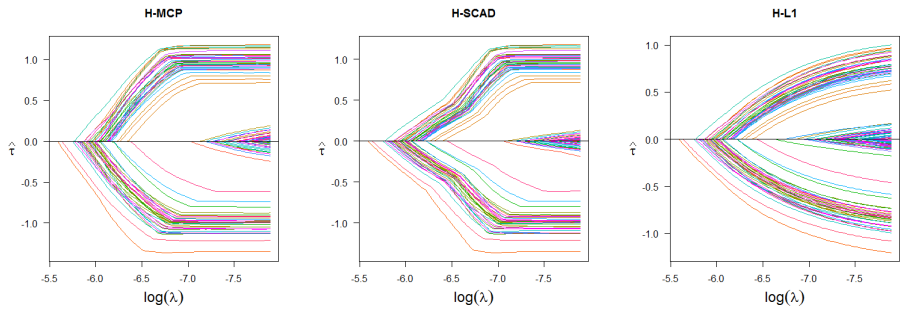


Figure 4.7: Solution paths of $\tau_i$'s by three HDRs on Example 3

Example 4

In this example, we consider the high-dimensional regression setting.

**Case 1.** We generate data from this model:

$$y_i = \tau_i + \beta_0 + \sum_{i=1}^{1000} x_{ij}^\top \beta_j + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $x_i = (x_{i1}, \ldots, x_{i1000})$'s and $\varepsilon_i$'s are generated using the same settings in Example 2, but $\beta = (\beta_0, \ldots, \beta_{1000})$ is sparse in the sense that only the $\beta_0, \ldots, \beta_5$ are nonzero. We set $\beta_0 = 1$ and generate $\beta_1, \ldots, \beta_5$ from Uniform(0.5, 1). For the generation of $\tau_i$'s, we consider combinations of two deviation amounts $\Gamma = 1, 10$ and two heterogeneity percentages 20%, 40%.

Since in this example $\beta$ is also high-dimensional, we penalizes $\beta$ in all the methods (except the intercept $\beta_0$) in addition to the penalization on $\tau$. For the three baseline methods, we impose $L_1$ penalties on $\beta$ as implemented by `hqreg`. For the HDR methods, $\tau$ and $\beta$ are always penalized with the same penalty function since `ncvreg` is used. The penalization on $\tau$ and $\beta$ can only differ in the regularization parameters. Recall the penalty term has the form of (4.3) with separate regularization parameters $\lambda_1$ and $\lambda_2$ for $\tau$ and $\beta$ respectively, and here we set $\lambda_1 = 0.02\lambda_2$. The rationale behind such setting comes from the fact the coordinate descent update of each $\tau_i$ is only associated with one observation $(x_i, y_i)$ for thresholding and when

$\lambda_1 = \lambda_2$ it is much easier for the algorithm to detect nonzero $\beta_j$'s than nonzero $\tau_i$'s. Hence in this situation, nonzero estimates $\widehat{\tau}_i$'s appear only near the end of the solution paths where the $\lambda$ value is too small to suppress noises and thus true signals and noises become indistinguishable. Using $\lambda_1 = 0.02\lambda_2$ will make the penalization on $\tau_i$'s lighter than that on $\beta_j$'s so that nonzero $\tau_i$'s can be discovered earlier in the paths. Figure 4.8 gives an intuitive illustration of this choice. In the first two cases where $\lambda_1 = \lambda_2$ or $\lambda_1 = 0.1\lambda_2$, the algorithm is able to discover the nonzero $\beta_1, \ldots, \beta_5$ while completely overlooking the nonzero $\tau_i$'s. In the third case where $\lambda_1 = 0.05\lambda_2$ and $\tau_i$'s are less penalized, nonzero $\tau_i$'s are detected while some noises arise in the mean time. And in the last case where $\lambda_1 = 0.02\lambda_2$ and the relative penalization on $\tau_i$'s are further reduced, nonzero $\tau_i$'s are detected even earlier and separated from the confounding noises.

Under such settings we carry out a series of simulations with 100 repetitions, the results of which are reported in Table 4.7. Since $\beta$ is sparse, we also list FDP($\beta$), MP($\beta$) in addition to the estimation biases and $L_2$ errors. Table 4.7 shows that H-MCP and H-SCAD are dominant in all four scenarios. They have very accurate variable selection as well as extremely small estimation errors. As for H-$L_1$, although the method has relatively small estimation errors, its estimates $\widehat{\beta}$ typically contain a very large proportion of false positives. The reason here is similar to what is shown in Figure 4.7: the MCP and SCAD penalties allow nonzero estimates to stabilize quickly to nearly unbiased levels while $L_1$ makes them develop slowly until eventually confounded by noises.

We can also learn from Table 4.7 how the performances vary across different scenarios. When the heterogeneity percentage changes from 20% to 40%, almost all methods perform worse, probably because the true coefficient vector $(\tau, \beta)$ become denser and harder to estimate with the given sample size. When the deviation amount $\Gamma$ increases from 1 to 10, the performances of HDRs do not seem to be

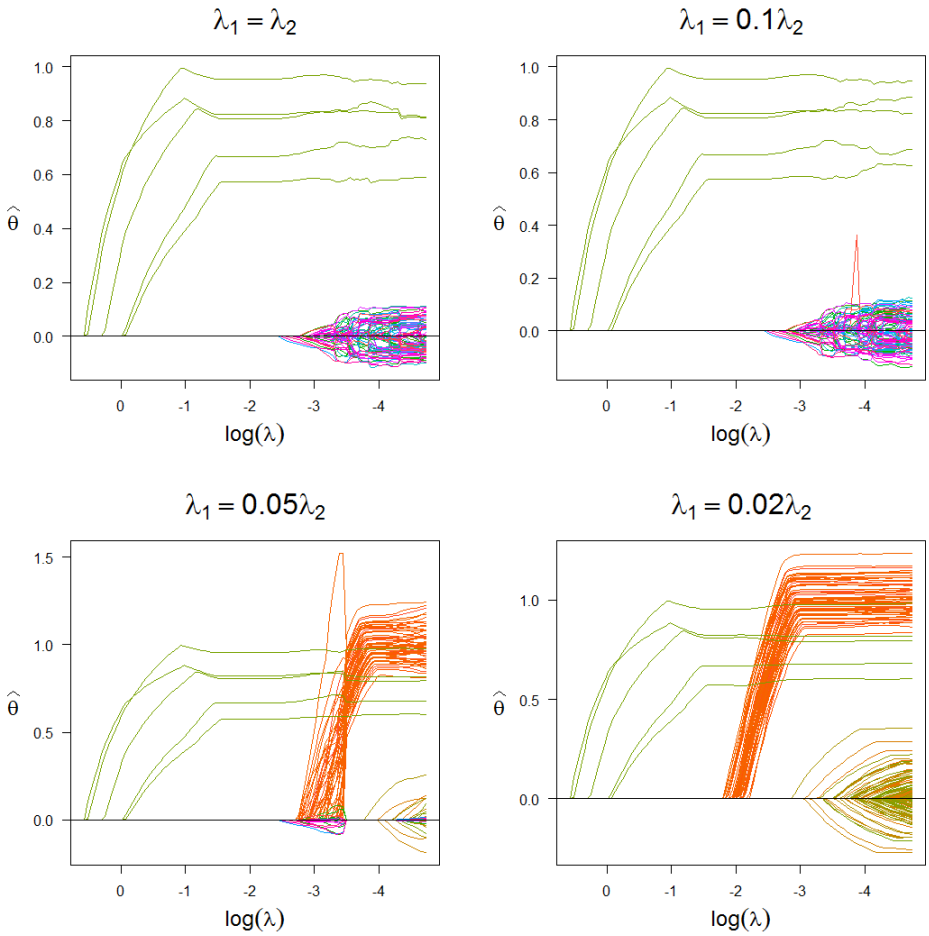


Figure 4.8: Solution paths fitted by H-MCP on Example 4 – Case 1 with deviation amount 1 and heterogeneity percentage 0.2. Different ratios between $\lambda_1$ and $\lambda_2$ are compared.

| | | Percentage of Heterogeneity | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 20% | | | | 40% | | | |
| $\Gamma$ | Method | $\widehat{\beta_0} - \beta_0$ | $\|\widehat{\beta} - \beta\|_2$ | FDP($\beta$) | MP($\beta$) | $\widehat{\beta_0} - \beta_0$ | $\|\widehat{\beta} - \beta\|_2$ | FDP($\beta$) | MP($\beta$) |
| 1 | H-MCP | -0.001 | 0.015 | 0 | 0 | -0.001 | 0.018 | 0 | 0 |
| | | (0.007) | (0.005) | (0) | (0) | (0.009) | (0.006) | (0) | (0) |
| | H-SCAD | 4e-4 | 0.015 | 0 | 0 | 2e-4 | 0.017 | 0 | 0 |
| | | (0.007) | (0.004) | (0) | (0) | (0.009) | (0.006) | (0) | (0) |
| | H-$L_1$ | 0.052 | 0.085 | 0.628 | 0 | 0.398 | 0.463 | 0.270 | 0 |
| | | (0.010) | (0.014) | (0.137) | (0) | (0.022) | (0.026) | (0.161) | (0) |
| | Huber | 0.095 | 0.151 | 0.374 | 0 | 0.337 | 0.426 | 0.714 | 0 |
| | | (0.015) | (0.024) | (0.177) | (0) | (0.025) | (0.036) | (0.083) | (0) |
| | LAD | 0.054 | 0.112 | 0.317 | 0 | 0.302 | 0.417 | 0.695 | 0 |
| | | (0.016) | (0.027) | (0.179) | (0) | (0.043) | (0.059) | (0.097) | (0) |
| | LS | 0.198 | 0.289 | 0.206 | 0 | 0.399 | 0.458 | 0.333 | 0 |
| | | (0.020) | (0.022) | (0.179) | (0) | (0.021) | (0.022) | (0.188) | (0) |
| 10 | H-MCP | -0.001 | 0.015 | 0 | 0 | 0.015 | 0.075 | 0.013 | 0.017 |
| | | (0.006) | (0.004) | (0) | (0) | (0.116) | (0.409) | (0.094) | (0.117) |
| | H-SCAD | -0.001 | 0.015 | 0 | 0 | 0.038 | 0.081 | 0.005 | 0.008 |
| | | (0.007) | (0.004) | (0) | (0) | (0.389) | (0.637) | (0.050) | (0.083) |
| | H-$L_1$ | 0.054 | 0.087 | 0.623 | 0 | 1.406 | 1.923 | 0.800 | 0.352 |
| | | (0.010) | (0.014) | (0.139) | (0) | (0.555) | (0.754) | (0.082) | (0.237) |
| | Huber | 0.277 | 0.576 | 0.049 | 0.072 | 2.452 | 3.052 | 0.180 | 0.757 |
| | | (0.208) | (0.430) | (0.087) | (0.187) | (0.467) | (0.720) | (0.303) | (0.180) |
| | LAD | 0.303 | 0.690 | 0.047 | 0.118 | 2.384 | 2.919 | 0.129 | 0.803 |
| | | (0.248) | (0.499) | (0.089) | (0.236) | (0.334) | (0.336) | (0.235) | (0.083) |
| | LS | 1.997 | 2.432 | 0.427 | 0.393 | 3.999 | 4.289 | 0.291 | 0.615 |
| | | (0.120) | (0.141) | (0.263) | (0.234) | (0.140) | (0.144) | (0.298) | (0.196) |

Table 4.7: Estimation and variable selection of $\beta$ for Example 4 – Case 1. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

affected much, while those of the baseline methods vary quite a bit. LS is clearly the most sensitive to $\Gamma$, whose performances worsen in both estimation and variable selection. The two alternatives Huber and LAD have less estimation biases, but their variable selection performances seem unstable and unreliable.

| | | Percentage of Heterogeneity | | | |
|---|---|---|---|---|---|
| | | 20% | | 40% | |
| $\Gamma$ | Method | FDP | MP | FDP | MP |
| 1 | H-MCP | 0.020 | 0 | 0.069 | 0 |
| | | (0.035) | (0) | (0.092) | (0) |
| | H-SCAD | 0.043 | 0 | 0.029 | 0 |
| | | (0.033) | (0) | (0.027) | (0) |
| | H-$L_1$ | 0.440 | 0 | 0.240 | 0.973 |
| | | (0.082) | (0) | (0.244) | (0.017) |
| 10 | H-MCP | 0.012 | 0 | 0.045 | 0 |
| | | (0.029) | (0) | (0.092) | (0) |
| | H-SCAD | 0.002 | 0 | 0.011 | 0.001 |
| | | (0.008) | (0) | (0.061) | (0.007) |
| | H-$L_1$ | 0.418 | 0 | 0.549 | 8e-5 |
| | | (0.072) | (0) | (0.028) | (0.001) |

Table 4.8: Heterogeneity discovery performances for Example 4 – Case 1. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

Let us take another look at the heterogeneity discovery performances of the HDRs reported in Table 4.8. We observe that H-MCP and H-SCAD maintain close to zero FDPs and MPs across all scenarios, while H-$L_1$ always has high FDPs. In the case where $\Gamma = 1$ and the heterogeneity percentage is 40%, the MP of H-$L_1$ shows that it almost completely miss all the true outliers.

**Case 2.** We generate data from this model:

$$y_i = \beta_0 + x_{i1}(\beta_1 + \tau_i) + \sum_{i=2}^{1000} x_{ij}^\top \beta_j + \varepsilon_i, \quad i = 1, \ldots, n,$$

where all the settings are the same as in Case 1 except that the source of heterogeneity is the predictive effect of $x_1$. Similar to Example 1 Case 2, we also add H-MIS to the list to be compared with the other methods.

| | | Percentage of Heterogeneity | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 20% | | | | 40% | | | |
| $\Gamma$ | Method | $\widehat{\beta_1} - \beta_1$ | $\|\widehat{\beta} - \beta\|_2$ | FDP($\beta$) | MP($\beta$) | $\widehat{\beta_1} - \beta_1$ | $\|\widehat{\beta} - \beta\|_2$ | FDP($\beta$) | MP($\beta$) |
| 1 | H-MIS | 0.002 | 0.016 | 0 | 0 | 0.013 | 0.030 | 0 | 0 |
| | | (0.007) | (0.005) | (0) | (0) | (0.100) | (0.098) | (0) | (0) |
| | H-MCP | 0.015 | 0.026 | 0 | 0 | 0.055 | 0.065 | 0.003 | 0 |
| | | (0.009) | (0.008) | (0) | (0) | (0.040) | (0.040) | (0.020) | (0) |
| | H-SCAD | 0.019 | 0.029 | 0.001 | 0 | 0.072 | 0.081 | 0.011 | 0 |
| | | (0.010) | (0.009) | (0.014) | (0) | (0.047) | (0.048) | (0.042) | (0) |
| | H-$L_1$ | 0.009 | 0.067 | 0.514 | 0 | 0.093 | 0.141 | 0.611 | 0 |
| | | (0.010) | (0.016) | (0.179) | (0) | (0.024) | (0.036) | (0.168) | (0) |
| | Huber | 0.024 | 0.083 | 0.242 | 0 | 0.125 | 0.182 | 0.443 | 0 |
| | | (0.013) | (0.019) | (0.163) | (0) | (0.035) | (0.040) | (0.196) | (0) |
| | LAD | -0.003 | 0.072 | 0.214 | 0 | 0.046 | 0.117 | 0.276 | 0 |
| | | (0.013) | (0.021) | (0.160) | (0) | (0.021) | (0.041) | (0.171) | (0) |
| | LS | 0.103 | 0.232 | 0.227 | 0 | 0.295 | 0.377 | 0.297 | 0 |
| | | (0.035) | (0.039) | (0.190) | (0) | (0.047) | (0.052) | (0.187) | (0) |
| 10 | H-MIS | -0.001 | 0.015 | 0 | 0 | -0.0004 | 0.019 | 0 | 0 |
| | | (0.006) | (0.004) | (0) | (0) | (0.007) | (0.006) | (0) | (0) |
| | H-MCP | 0.032 | 0.082 | 0.001 | 0 | 0.134 | 0.224 | 0.014 | 0.008 |
| | | (0.034) | (0.065) | (0.014) | (0) | (0.132) | (0.212) | (0.048) | (0.037) |
| | H-SCAD | 0.044 | 0.109 | 0.003 | 0 | 0.204 | 0.332 | 0.102 | 0.003 |
| | | (0.044) | (0.099) | (0.025) | (0) | (0.200) | (0.305) | (0.141) | (0.023) |
| | H-$L_1$ | 0.050 | 0.275 | 0.639 | 0 | 0.479 | 0.778 | 0.735 | 0.008 |
| | | (0.034) | (0.090) | (0.139) | (0) | (0.156) | (0.229) | (0.078) | (0.037) |
| | Huber | 0.015 | 0.143 | 0.023 | 0 | 0.210 | 0.581 | 0.065 | 0.088 |
| | | (0.020) | (0.052) | (0.063) | (0) | (0.111) | (0.413) | (0.092) | (0.195) |
| | LAD | -0.011 | 0.120 | 0.043 | 0 | 0.147 | 0.504 | 0.066 | 0.097 |
| | | (0.018) | (0.060) | (0.080) | (0) | (0.154) | (0.503) | (0.100) | (0.197) |
| | LS | 1.257 | 1.847 | 0.427 | 0.362 | 3.092 | 3.431 | 0.370 | 0.490 |
| | | (0.308) | (0.276) | (0.242) | (0.198) | (0.443) | (0.421) | (0.261) | (0.179) |

Table 4.9: Estimation and variable selection of $\beta$ for Example 4 – Case 2. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

| | | Percentage of Heterogeneity | | | |
|---|---|---|---|---|---|
| | | 20% | | 40% | |
| $\Gamma$ | Method | FDP | MP | FDP | MP |
| 1 | H-MIS | 0.113 | 0.160 | 0.143 | 0.127 |
| | | (0.035) | (0.052) | (0.100) | (0.078) |
| | H-MCP | 0.203 | 0.370 | 0.163 | 0.380 |
| | | (0.035) | (0.073) | (0.050) | (0.084) |
| | H-SCAD | 0.224 | 0.371 | 0.199 | 0.385 |
| | | (0.033) | (0.075) | (0.044) | (0.089) |
| | H-$L_1$ | 0.273 | 0.368 | 0.204 | 0.412 |
| | | (0.082) | (0.069) | (0.054) | (0.083) |
| 1 | H-MIS | 0.004 | 0.025 | 0.001 | 0.028 |
| | | (0.011) | (0.019) | (0.002) | (0.014) |
| | H-MCP | 0.007 | 0.226 | 0.029 | 0.237 |
| | | (0.014) | (0.062) | (0.034) | (0.054) |
| | H-SCAD | 0.014 | 0.226 | 0.061 | 0.239 |
| | | (0.025) | (0.062) | (0.052) | (0.056) |
| | H-$L_1$ | 0.099 | 0.231 | 0.234 | 0.257 |
| | | (0.053) | (0.061) | (0.047) | (0.052) |

Table 4.10: Heterogeneity discovery performances for Example 4 – Case 2. Mean values and standard errors (shown in parentheses) based on 100 repetitions.

Table 4.9 presents the estimation performances for this case. Generally speaking, the patterns are similar to Case 1, but one noticeable difference is that Huber and LAD shows much more robustness in Case 2 than in Case 1. From Table 4.9, we observe that Huber and LAD can actually have similar estimation performances as H-MCP or H-SCAD, although they seem less accurate in variable selection. It is also noteworthy that H-MIS has in fact better overall performances than H-MCP and H-SCAD. Table 4.10 reports the heterogeneity discovery performances and shows that H-MIS has the smallest FDPs and MPs in general. Compared with Case 1, here H-MCP and H-SCAD seem to be less accurate, especially in the two scenarios that the deviation amount $\Gamma = 1$ and heterogeneity is harder to detect than when $\Gamma = 10$.

## 4.6 Building energy efficiency data example

In this section, we use the building energy efficiency dataset to illustrate the HDR methods. This dataset is available at UCI machine learning repository and initially contributed by Tsanas and Xifara (2012). It contains 768 residential buildings generated by Ecotest, an advanced building energy simulation tool, which are characterized by eight building parameters: Relative Compactness (RC), Surface Area (SA), Wall Area (WA), Roof Area (RA), Overall Height (H), Orientation (O), Glazing Area (GA), Glazing Area Distribution (GAD). Then the heating load (HL) and cooling load (CL) for each building was recorded, which are useful for determining the specifications of heating and cooling equipment required to maintain comfortable indoor temperature. Hence in order to design energy-efficient buildings, it will be interesting to study how the building parameters affect HL and CL and whether there is a hidden heterogeneity pattern in their relationship that may indicate influences of latent factors. In this example, we perform regression analysis with HL as the output. Due to the data generation settings, there are two pairs of

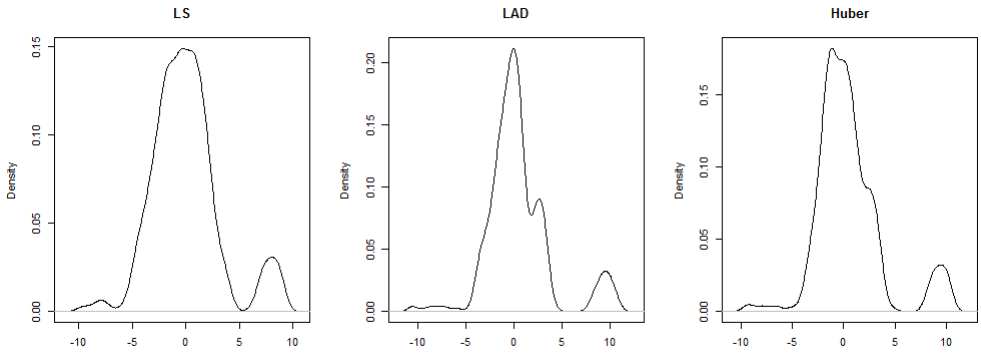


Figure 4.9: Density plots of the residuals computed by LS, LAD and Huber loss regression

building parameters that are almost inversely proportional: RC and SA, RA and H. Hence we exclude SA and H, and use the rest as input variables.

We first consider the three baseline methods, LS, LAD and Huber loss regression, and plot the kernel density estimates of the computed residuals in Figure 4.9. We could clearly see that after adjusting for the input variables, the distributions of residuals for all three methods still appear multimodal. This heterogeneity patterns may be caused by some unobserved factors, and to investigate that we need to first identify the heterogeneous observations.

Hence it is not suitable to fit a standard regression model with a common intercept, and instead we fit a heterogeneous model $y = \beta_0 + \tau_i + \sum_{j=1}^{6} x_{ij}\beta_j$ with HDR methods, H-MCP, H-SCAD and H-$L_1$. Here we only penalize the deviation effects $\tau_i$'s and select the tuning parameter $\lambda$ by minimizing modified BIC as in previous simulation studies. In each case, the method separates the observations into two subgroups, the majority group with $\widehat{\tau}_i = 0$ and a homogeneous pattern, and the deviation group with various heterogeneity patterns corresponding to nonzero $\widehat{\tau}_i$'s. From Table 4.11, we observe that for each method the size of the deviation group is less than 12% of the entire dataset, so there is a serious imbalance between the two subgroups.

| Method | Size of deviation group |
|---|---|
| H-MCP | 85 |
| H-SCAD | 92 |
| H-$L_1$ | 88 |

Table 4.11: Heterogeneity discovery for the building energy dataset

How does heterogeneity discovery affect the distribution of residuals? We plot the residuals $y - \widehat{\beta}_0 - \sum_{j=1}^{6} x_{ij}\widehat{\beta}_j$ (not including $\widehat{\tau}_i$'s) in each subgroup for the H-MCP method in Figure 4.10. We observe that with the removal of relatively few heterogeneous observations, the residuals for majority group appear much more homogeneous, and instead of spanning from -10 to 10 the distribution is concentrated between -6 and 6. On the other hand, the distribution of the deviation group appear ragged and multimodal, whose irregularity and heterogeneity is captured by nonzero deviation $\tau_i$'s after adjusting for the common intercept. The density plots for H-SCAD and H-$L_1$ in Figure 4.11, 4.12 show similar patterns.

Table 4.12 reports the corresponding coefficient estimates for the input variables. HDR methods and the two robust regression methods Huber loss regression and LAD yield similar estimates, but they are largely different from the estimates of LS, especially the intercept and the coefficient for RC.

In order to better assess the reliability of the estimation of the coefficients, we also generate 50 artificial samples from the dataset via bootstrapping and apply the methods to each sample. Then we record the means and the standard errors of the coefficient estimates as shown in Table 4.13. The mean values are similar to the coefficient estimates for the original data for all methods, but the standard errors differ. Bootstrapping causes the underlying heterogeneity patterns to vary mildly across samples, and LS is clearly sensitive to such perturbation. On the

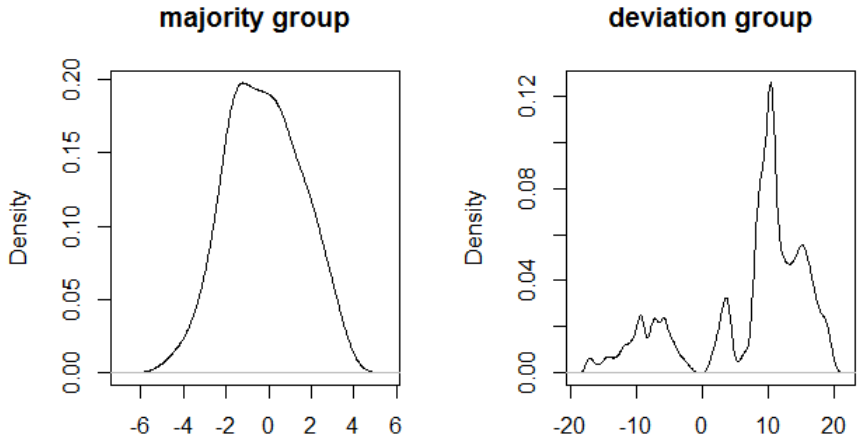


Figure 4.10: Density plots of the residuals (adjusted with the common intercept $\widehat{\beta}_0$ but not $\widehat{\tau}_i$'s) in each subgroup identified by H-MCP

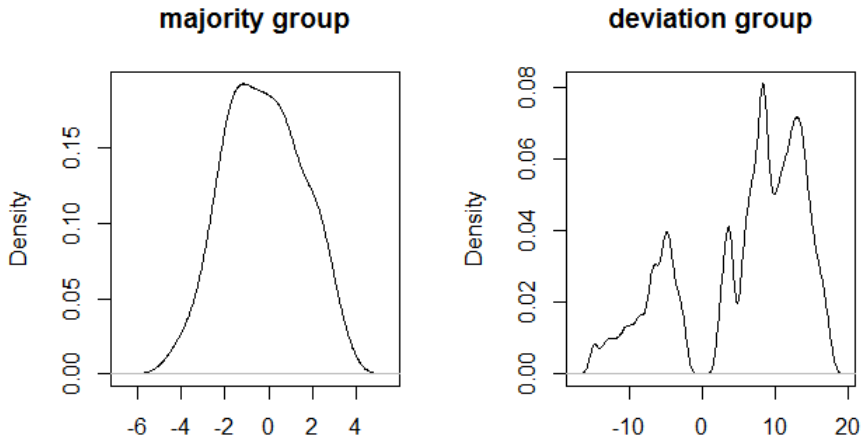


Figure 4.11: Density plots of the residuals (adjusted with the common intercept $\widehat{\beta}_0$ but not $\widehat{\tau}_i$'s) in each subgroup identified by H-SCAD

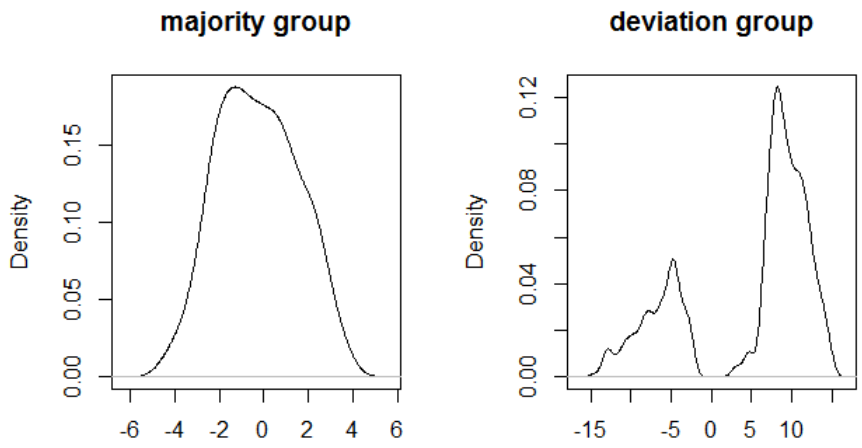


Figure 4.12: Density plots of the residuals (adjusted with the common intercept $\widehat{\beta}_0$ but not $\widehat{\tau}_i$'s) in each subgroup identified by H-$L_1$

| Method | Intercept | RC | WA | RA | O | GA | GAD |
|---|---|---|---|---|---|---|---|
| H-MCP | 161.372 | -81.422 | -0.054 | -0.366 | -0.021 | 18.111 | 0.062 |
| H-SCAD | 185.273 | -96.858 | -0.071 | -0.404 | -0.024 | 18.535 | 0.101 |
| H-$L_1$ | 209.974 | -112.737 | -0.089 | -0.443 | -0.024 | 18.825 | 0.118 |
| Huber | 187.727 | -98.034 | -0.073 | -0.409 | -0.009 | 18.120 | 0.121 |
| LAD | 180.377 | -92.175 | -0.068 | -0.398 | 0.006 | 15.843 | 0.068 |
| LS | 258.152 | -143.895 | -0.123 | -0.520 | -0.023 | 19.933 | 0.204 |

Table 4.12: Coefficient estimates for the building energy dataset

other hand, we observe that the estimates by H-MCP and H-SCAD show much less variability compared to LS. For each coefficient, the standard error of H-MCP or H-SCAD is 30% to 50% smaller than that of LS. However, H-$L_1$ does not perform as well as these two. It actually has the largest variability, which again indicates the disadvantage of using $L_1$ penalty instead of nonconvex ones for HDR. Finally, Huber loss regression and LAD regression also show better stability than LS, only outperformed by H-MCP and H-SCAD.

| | | Intercept | RC | WA | RA | O | GA | GAD |
|---|---|---|---|---|---|---|---|---|
| H-MCP | Mean | 166.459 | -84.686 | -0.058 | -0.374 | -0.017 | 18.259 | 0.070 |
| | SE | 8.541 | 5.523 | 0.007 | 0.013 | 0.074 | 0.758 | 0.060 |
| H-SCAD | Mean | 184.570 | -96.397 | -0.071 | -0.403 | -0.016 | 18.576 | 0.106 |
| | SE | 9.223 | 5.856 | 0.007 | 0.014 | 0.083 | 0.784 | 0.065 |
| H-$L_1$ | Mean | 213.285 | -114.909 | -0.091 | -0.449 | -0.012 | 18.907 | 0.130 |
| | SE | 17.346 | 11.094 | 0.012 | 0.028 | 0.091 | 0.810 | 0.068 |
| Huber | Mean | 188.668 | -98.640 | -0.074 | -0.410 | 0.003 | 18.134 | 0.126 |
| | SE | 12.121 | 7.749 | 0.009 | 0.019 | 0.090 | 0.879 | 0.066 |
| LAD | Mean | 182.949 | -94.053 | -0.070 | -0.402 | 0.013 | 16.183 | 0.081 |
| | SE | 11.863 | 7.700 | 0.008 | 0.019 | 0.078 | 1.058 | 0.076 |
| LS | Mean | 257.881 | -143.764 | -0.123 | -0.520 | -0.009 | 19.885 | 0.214 |
| | SE | 14.471 | 9.260 | 0.010 | 0.023 | 0.119 | 1.005 | 0.086 |

Table 4.13: Means and standard errors (SE) of coefficient estimates on 50 bootstrap samples from the building energy dataset

# CHAPTER 5
# SUMMARY AND DISCUSSION

This thesis studies two different approaches for dealing with data heterogeneity in the potentially high dimensional settings. In Chapter 3, we consider two robust regression methods, the Huber loss regression and the quantile regression. They have important applications in many fields, but there is a lack of efficient algorithms and publicly available implementation that works well for the high-dimensional case. We develop an efficient and scalable algorithm, Semismooth Newton Coordinate Descent (SNCD), for computing the solution paths of these models with the elastic-net penalty. We also provide an implementation via the R package `hqreg` publicly available on CRAN (`http://cloud.r-project.org/package=hqreg`). In Chapter 4, we propose a nonconvex penalized regression method, Heterogeneity Discovery Regression (HDR), for simultaneously detecting heterogeneity and estimating regression coefficients. We establish the statistical precision for any local optimizer of its objective and demonstrate significant advantages of using HDR over alternatives like robust regression through extensive numerical experiments.

For future work, we need to address some theoretical issues and consider possible extensions. For SNCD, it is of interest to further investigate its convergence rate more formally. Also, it is worthwhile to extend the algorithm to other forms of regularizers, including nonconvex penalties such as MCP, SCAD and group penalties such as group-lasso. For HDR, it will be interesting to consider other loss functions such as Huber loss and LAD loss and derive similar corollaries that provide probabilistic guarantees for the corresponding estimates under our theoretical framework. We can also extend the approach to deal with heterogeneity in panel data. In that case, since the data contains repeated measurements for each subject, both coefficient estimation heterogeneity detection will be more reliable.

# APPENDIX A
# SNA FOR PENALIZED HUBER LOSS REGRESSION

## A.1 Derivation

Following section 2.2, denote $\mathcal{S}(z) = (S(z_1), \ldots, S(z_p))^\top$ and

$$d(\beta_0, \beta) = (h'_\gamma(y_1 - \beta_0 - x_1^\top \beta), \ldots, h'_\gamma(y_n - \beta_0 - x_n^\top \beta))^\top,$$

then the KKT conditions (3.6) can be written as (3.14).

Since the soft-thresholding operator is piecewise linear as shown in (3.11), we define

$$\begin{aligned} A &= \{j : |\beta_j + s_j| > 1\}, \\ B &= \{j : |\beta_j + s_j| \leq 1\}. \end{aligned} \tag{A.1}$$

The set $A$ works as an estimate for the support of $\beta$. In fact, if $(\widehat{s}, \widehat{\beta}_0, \widehat{\beta})$ satisfies the KKT conditions, then the set $A$ defined on $(\widehat{\beta}, \widehat{s})$ is exactly the support for $\widehat{\beta}$. This is easy to see: since $\widehat{s}_j \in \partial|\widehat{\beta}_j|$, if $\widehat{\beta}_j \neq 0$ then $|\widehat{\beta}_j + \widehat{s}_j| = |\widehat{\beta}_j + \operatorname{sgn}(\widehat{\beta}_j)| = |\widehat{\beta}_j| + 1 > 1$, otherwise $|\widehat{\beta}_j + \widehat{s}_j| = |\widehat{s}_j| \leq 1$.

We decompose $\beta$ into $\beta_A$, $\beta_B$ and $s$ into $s_A$, $s_B$, and denote $Z = (s_A^\top, \beta_B^\top, \beta_0, \beta_A^\top, s_B^\top)^\top$. Then KKT conditions (3.14) can be rewritten as

$$F(Z) = \begin{bmatrix} \beta_A - \mathcal{S}(\beta_A + s_A) \\ \beta_B - \mathcal{S}(\beta_B + s_B) \\ -\frac{1}{n} 1^\top d \\ -\frac{1}{n} X_A^\top d + \lambda\alpha s_A + \lambda(1-\alpha)\beta_A \\ -\frac{1}{n} X_B^\top d + \lambda\alpha s_B + \lambda(1-\alpha)\beta_B \end{bmatrix} = \mathbf{0}. \tag{A.2}$$

And from (3.11) we have

$$\begin{cases} \beta_A - \mathcal{S}(\beta_A + s_A) = -s_A + \operatorname{sgn}(\beta_A + s_A), \\ \beta_B - \mathcal{S}(\beta_B + s_B) = \beta_B. \end{cases} \tag{A.3}$$

Let $\psi_\gamma$ be as in (3.10), and for brevity denote $\psi_i = \psi_\gamma(\, y_i - \beta_0 - x_i^\top \beta)$, and

$\Psi = \Psi(\beta_0, \beta) = \frac{1}{n}\text{diag}(\psi_1, \ldots, \psi_n)$. Then the following result gives a proper Newton derivative of $F(Z)$.

**Theorem A.1.** *$F(Z)$ is Newton differentiable for any $Z \in \mathbb{R}^{2p+1}$ and*

$$H(Z) \coloneqq \begin{bmatrix} -I_{|A|} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{|B|} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n^\top \Psi X_B & \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A & \mathbf{0} \\ \lambda\alpha I_{|A|} & X_A^\top \Psi X_B & X_A^\top \Psi \mathbf{1}_n & X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|} & \mathbf{0} \\ \mathbf{0} & X_B^\top \Psi X_B + \lambda(1-\alpha) I_{|B|} & X_B^\top \Psi \mathbf{1}_n & X_B^\top \Psi X_A & \lambda\alpha I_{|B|} \end{bmatrix} \in \nabla_N F(Z).$$

*Furthermore, for any $\gamma > 0$ and $\alpha \in (0,1)$, on the set $\{Z = (s, \beta_0, \beta) : \text{ there exists } i \in \{1, \ldots, n\} \text{ such that } |y_i - \beta_0 - x_i^\top \beta| \le \gamma\}$, $H(Z)$ is invertible and $H(Z)^{-1}$ is uniformly bounded in spectral norm.*

From Theorems 2.6 and A.1, we immediately obtain the following result.

**Theorem A.2.** *Given $\lambda, \gamma, \alpha \in (0,1)$, define $Z$ and $F(Z)$ as* (A.2)*. Suppose $\widehat{Z}$ solves $F(Z) = 0$ and there exists a neighborhood $\mathcal{N}(\widehat{Z})$ such that for any $Z \in \mathcal{N}(\widehat{Z})$ there is an $i \in \{1, \ldots, n\}$ that satisfies $|y_i - \beta_0 - x_i^\top \beta| \le \gamma$, then the Newton-type iteration*

$$Z^{k+1} = Z^k - H(Z^k)^{-1} F(Z^k)$$

*converges superlinearly to $\widehat{Z}$ provided that $\|Z^0 - \widehat{Z}\|_2$ is sufficiently small.*

Now we describe the algorithm in details. The $(k+1)$-th iteration can be split into two steps:

1. Solve $D_k$ from $H(Z^k) D_k = -F(Z^k)$;
2. Update $Z^{k+1} = Z^k + D^k$.

At the first glance, step 1 seems to involve inverting a $(2p+1) \times (2p+1)$ matrix, which is intractable in high dimensional settings. However, the definitions of sets $A, B$ in (A.1) motivate an "active set strategy" for dimension reduction. Given the estimates from the $k$th iteration, define the active set $A_k$ and its complement $B_k$ by (3.15), $d_k = d(\beta_0^k, \beta^k)$, and $D_k = (D_{A_k}^{s\top}, D_{B_k}^{\beta\top}, D_0^{\beta_0}, D_{A_k}^{\beta\top}, D_{B_k}^{s\top})^\top$ corresponding to $Z_k$.

Now substituting these identities into step 1 and combining (A.3) we have

$$
\begin{aligned}
D^s_{A_k} &= -s^k_{A_k} + \mathrm{sgn}(\beta^k_{A_k} + s^k_{A_k}), \\
D^\beta_{B_k} &= -\beta^k_{B_k}, \\
\begin{bmatrix} D^{\beta_0}_0 \\ D^\beta_{A_k} \end{bmatrix} &= \begin{bmatrix} \mathbf{1}_n^\top \Psi_k \mathbf{1}_n & \mathbf{1}_n^\top \Psi_k X_{A_k} \\ X^\top_{A_k} \Psi_k \mathbf{1}_n & X^\top_{A_k} \Psi_k X_{A_k} + \lambda(1-\alpha) I_{|A_k|} \end{bmatrix}^{-1} \\
&\qquad \begin{bmatrix} \frac{1}{n} 1^\top d_k + \mathbf{1}_n^\top \Psi_k X_{B_k} \beta^k_{B_k} \\ \frac{1}{n} X^\top_{A_k} d_k - \lambda(1-\alpha)\beta^k_{A_k} - \lambda\alpha \mathrm{sgn}(\beta^k_{A_k} + s^k_{A_k}) + X^\top_{A_k} \Psi_k X_{B_k} \beta^k_{B_k} \end{bmatrix}, \\
D^s_{B_k} &= -s^k_{B_k} + \frac{1}{\lambda\alpha} X^\top_{B_k} \left( \frac{1}{n} d_k + \Psi_k X_{B_k} \beta^k_{B_k} - \Psi_k \mathbf{1}_n D^{\beta_0}_0 + \Psi_k X_{A_k} D^\beta_{A_k} \right).
\end{aligned}
$$

Combining steps 1 and 2, the $(k+1)$th iteration of SNA is carried out as follows:

(i) Update $s^{k+1}_{A_k}$ and $\beta^{k+1}_{B_k}$:

$$
\begin{aligned}
s^{k+1}_{A_k} &= \mathrm{sgn}(\beta^k_{A_k} + s^k_{A_k}), \\
\beta^{k+1}_{B_k} &= \mathbf{0}.
\end{aligned}
$$

(ii) Find the direction $D^{\beta_0}_0$ for the intercept $\beta_0$, and $D^\beta_{A_k}$ for the active coefficients $\beta_{A_k}$:

$$
\begin{aligned}
\begin{bmatrix} D^{\beta_0}_0 \\ D^\beta_{A_k} \end{bmatrix} &= \begin{bmatrix} \mathbf{1}_n^\top \Psi_k \mathbf{1}_n & \mathbf{1}_n^\top \Psi_k X_{A_k} \\ X^\top_{A_k} \Psi_k \mathbf{1}_n & X^\top_{A_k} \Psi_k X_{A_k} + \lambda(1-\alpha) I_{|A_k|} \end{bmatrix}^{-1} \\
&\qquad \begin{bmatrix} \frac{1}{n} 1^\top d_k + \mathbf{1}_n^\top \Psi_k X_{B_k} \beta^k_{B_k} \\ \frac{1}{n} X^\top_{A_k} d_k - \lambda(1-\alpha)\beta^k_{A_k} - \lambda\alpha s^{k+1}_{A_k} + X^\top_{A_k} \Psi_k X_{B_k} \beta^k_{B_k} \end{bmatrix}.
\end{aligned}
$$

(iii) Update the intercept, the active coefficients, and the inactive subgradients:

$$
\begin{aligned}
\beta^{k+1}_0 &= \beta^k_0 + D^{\beta_0}_0, \\
\beta^{k+1}_{A_k} &= \beta^k_{A_k} + D^\beta_{A_k}, \\
s^{k+1}_{B_k} &= \frac{1}{\lambda\alpha} X^\top_{B_k} \left( \frac{1}{n} d_k + \Psi_k X_{B_k} \beta^k_{B_k} - \Psi_k \mathbf{1}_n D^{\beta_0}_0 + \Psi_k X_{A_k} D^\beta_{A_k} \right).
\end{aligned}
$$

## A.2 Proofs

Here we first give the proof of Theorem A.1. A new lemma is used in this proof, so it will be presented and proved subsequently.

*Proof.* Notice $\mathcal{S}$ is piecewise-smooth, then by Lemma 2.3, 2.5 and Lemma 2.4 (iv) $F_1(Z)$ is Newton differentiable, and with (A.3)

$$\begin{bmatrix} -I_{|A|} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{|B|} & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \nabla_N F_1(Z).$$

Similarly, the Huber loss is also piecewise-smooth, and by Lemma 2.3, 2.5 and Lemma 2.4 (ii)-(iv), we have $F_2(Z)$ and $F_3(Z)$ are Newton differentiable and

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}_n^\top \Psi X_B & \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A & \mathbf{0} \end{bmatrix} \in \nabla_N F_2(Z),$$

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}_n^\top \Psi X_B & \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A & \mathbf{0} \\ \lambda\alpha I_{|A|} & X_A^\top \Psi X_B & X_A^\top \Psi \mathbf{1}_n & X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|} & \mathbf{0} \\ \mathbf{0} & X_B^\top \Psi X_B + \lambda(1-\alpha) I_{|B|} & X_B^\top \Psi \mathbf{1}_n & X_B^\top \Psi X_A & \lambda\alpha I_{|B|} \end{bmatrix} \in \nabla_N F_3(Z).$$

Again, by Lemma 2.4 (iv), $F(Z)$ is Newton differentiable and

$$H(Z) = \begin{bmatrix} -I_{|A|} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{|B|} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n^\top \Psi X_B & \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A & \mathbf{0} \\ \lambda\alpha I_{|A|} & X_A^\top \Psi X_B & X_A^\top \Psi \mathbf{1}_n & X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|} & \mathbf{0} \\ \mathbf{0} & X_B^\top \Psi X_B + \lambda(1-\alpha) I_{|B|} & X_B^\top \Psi \mathbf{1}_n & X_B^\top \Psi X_A & \lambda\alpha I_{|B|} \end{bmatrix} \in \nabla_N F(Z).$$

For brevity, write $H(Z)$ as $H$. Now we partition $H$ as follows:

$$H_1 = \begin{bmatrix} -I_{|A|} & \mathbf{0} \\ \mathbf{0} & I_{|B|} \end{bmatrix}, \; H_2 = \begin{bmatrix} \mathbf{0} & \mathbf{1}_n^\top \Psi X_B \\ \lambda\alpha I_{|A|} & X_A^\top \Psi X_B \\ \mathbf{0} & X_B^\top \Psi X_B + \lambda(1-\alpha) I_{|B|} \end{bmatrix},$$

$$H_3 = \begin{bmatrix} \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A & \mathbf{0} \\ X_A^\top \Psi \mathbf{1}_n & X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|} & \mathbf{0} \\ X_B^\top \Psi \mathbf{1}_n & X_B^\top \Psi X_A & \lambda\alpha I_{|B|} \end{bmatrix}. \tag{A.4}$$

Then it is clear that $H_1$ is invertible. Now if $H_3$ is also invertible, which we show in Lemma A.3 under a mild condition, then via some algebra we have

$$H^{-1} = \begin{bmatrix} H_1^{-1} & \mathbf{0} \\ -H_3^{-1} H_2 H_1^{-1} & H_3^{-1} \end{bmatrix}. \tag{A.5}$$

Let $g = (g_1^\top, g_2^\top)^\top \in \mathbb{R}^p \times \mathbb{R}^{p+1}$, then

$$\begin{aligned} \|H^{-1} g\|_2^2 &= \|H_1^{-1} g_1\|_2^2 + \| - H_3^{-1} H_2 H_1^{-1} g_1 + H_3^{-1} g_2\|_2^2 \\ &\le \|H_1^{-1}\|^2 \|g_1\|_2^2 + (\|H_3^{-1}\| \|H_2\| \|H_1^{-1}\| \|g_1\|_2 + \|H_3^{-1}\| \|g_2\|_2)^2 \\ &\le (\|H_1^{-1}\| \|g_1\|_2 + \|H_3^{-1}\| \|H_2\| \|H_1^{-1}\| \|g_1\|_2 + \|H_3^{-1}\| \|g_2\|_2)^2 \\ &\le (\|H_1^{-1}\| + \|H_3^{-1}\| + \|H_3^{-1}\| \|H_2\| \|H_1^{-1}\|)^2 \|g\|_2^2 \end{aligned} \tag{A.6}$$

which implies

$$\|H^{-1}\| \le \|H_1^{-1}\| + \|H_3^{-1}\| + \|H_3^{-1}\| \|H_2\| \|H_1^{-1}\|. \tag{A.7}$$

Notice $\|X_A\| \vee \|X_B\| \le \|X\|$. Take $X_A$, without loss of generality shuffle columns of $X$ such that $X = \begin{pmatrix} X_A & X_B \end{pmatrix}$, then for any $g \in \mathbb{R}^{|A|}$ such that $\|g\|_2 = 1$, we have

$$\|X_A g\|_2 = \|X \begin{pmatrix} g \\ \mathbf{0} \end{pmatrix} \|_2 \le \sup \{\|Xv\|_2 : \; \|v\|_2 = 1\} = \|X\|,$$

implying that $\|X_A\| = \sup \{\|X_A g\|_2 : \; \|g\|_2 = 1\} \le \|X\|$. Similarly for $X_B$. Then a similar argument as in (A.6) shows that

$$\|H_2\| \le 1 + \alpha + 2\|X\|^2. \tag{A.8}$$

Combining (A.7), (A.8) with results of Lemma A.3 under its condition, and observing that $\|H_1^{-1}\| = 1$, we obtain the uniform boundedness of $H$ in spectral norm,

i.e.,

$$\begin{aligned}\|H^{-1}\| \;\leq\; & 1+\left[\frac{1}{\lambda\alpha}+\left(\frac{1}{\lambda(1-\alpha)}+\frac{\lambda_{\max}(X^\top X)^2+n\gamma\lambda(1-\alpha)}{\lambda(1-\alpha)}\left(1+\frac{\|X\|}{\sqrt{n}\gamma\lambda(1-\alpha)}\right)^2\right)\right. \\ & \left.\times\left(1+\frac{2\|X\|}{\sqrt{n}\gamma\lambda\alpha}\right)\right](2+\alpha+2\|X\|^2).\end{aligned}$$

□

The above proof has used the following technical lemma:

**Lemma A.3.** *Given $\alpha\in(0,1)$ and $\beta_0,\beta$ satisfy $|y_i-\beta_0-x_i^\top\beta|\leq\gamma$ for some $i$, then $H_3$ in* (A.4) *is invertible with its inverse uniformly bounded in spectral norm, i.e.*

$$\|H_3^{-1}\|\leq\frac{1}{\lambda\alpha}+\left[\frac{1}{\lambda(1-\alpha)}+\frac{\lambda_{\max}(X^\top X)^2+n\gamma\lambda(1-\alpha)}{\lambda(1-\alpha)}\left(1+\frac{\|X\|}{\sqrt{n}\gamma\lambda(1-\alpha)}\right)^2\right]\left(1+\frac{2\|X\|}{\sqrt{n}\gamma\lambda\alpha}\right).$$

*Proof.* Denote $J=n\gamma\Psi$, then $J$ is diagonal and idempotent. We have

$$\mathbf{1}_n^\top\Psi\mathbf{1}_n=\frac{1}{n\gamma}\mathbf{1}_n^\top J\mathbf{1}_n=\frac{1}{n\gamma}(J\mathbf{1}_n)^\top(J\mathbf{1}_n),$$

and

$$\begin{aligned}&\mathbf{1}_n^\top\Psi X_A\left(X_A^\top\Psi X_A+\lambda(1-\alpha)I_{|A|}\right)^{-1}X_A^\top\Psi\mathbf{1}_n \\ &=\;\frac{1}{n\gamma}(J\mathbf{1}_n)^\top(JX_A)\left((JX_A)^\top(JX_A)+n\gamma\lambda(1-\alpha)I_{|A|}\right)^{-1}(JX_A)^\top(J\mathbf{1}_n).\end{aligned}$$

Denote $a=J\mathbf{1}_n$, $Z=JX_A$, $t=n\gamma\lambda(1-\alpha)$, and $m=|A|$. Then the LHS becomes

$$\frac{1}{n\gamma}\left(a^\top a-a^\top Z(Z^\top Z+tI_m)^{-1}Z^\top a\right).$$

Since $|y_i-\beta_0-x_i^\top\beta|\leq\gamma$ for some $i$, we have $\psi_i=\frac{1}{n\gamma}>0$, implying that $J_{ii}=1$ and $a^\top a\geq J_{ii}^2=1$. Thus we are guaranteed that $a=J\mathbf{1}_n$ is not a zero vector.

Now apply SVD to $Z$ such that $Z=UDV^\top$, where $U_{n\times n}$ and $V_{m\times m}$ are both orthogonal matrices, and $D_{n\times m}$ is a rectangular diagonal matrix with non-negative

diagonal elements $d_1, \ldots, d_{m \wedge n}$. Hence

$$\begin{aligned} Z(Z^\top Z + tI_m)^{-1} Z^\top &= UDV^\top (VD^\top U^\top UDV^\top + tI_m)^{-1} VD^\top U^\top \\ &= UDV^\top \left(V(D^\top D + tI_m)V^\top\right)^{-1} VD^\top U^\top \\ &= UDV^\top V(D^\top D + tI_m)^{-1} V^\top VD^\top U^\top \\ &= UD(D^\top D + tI_m)^{-1} D^\top U^\top. \end{aligned}$$

When $n > m$,

$$D(D^\top D + tI_m)^{-1} D^\top = \mathrm{diag}(\frac{d_1^2}{d_1^2 + t}, \ldots, \frac{d_m^2}{d_m^2 + t}, 0, \ldots, 0),$$

and when $n \leq m$,

$$D(D^\top D + tI_m)^{-1} D^\top = \mathrm{diag}(\frac{d_1^2}{d_1^2 + t}, \ldots, \frac{d_n^2}{d_n^2 + t}).$$

In either case $D(D^\top D + tI_m)^{-1} D^\top$ is p.s.d. with $\lambda_{\max}(D(D^\top D + tI_m)^{-1} D^\top) < 1$.

Next we will derive the upper bound of eigenvalues of the above matrix. First, for any eigenvalue $d$ and corresponding nonzero eigenvector $u$ of $Z^\top Z = X_A J X_A$, we have

$$du^\top u = u^\top X_A^\top J X_A u = \begin{bmatrix} u \\ 0 \end{bmatrix}^\top X^\top JX \begin{bmatrix} u \\ 0 \end{bmatrix} \leq \lambda_{\max}(X^\top JX) u^\top u,$$

hence $d \leq \lambda_{\max}(X^\top JX)$. Then again, for any eigenvalue $c$ and corresponding nonzero eigenvector $v$ of $X^\top JX$, we have

$$cv^\top v = v^\top X^\top JXv = \sum_i J_{ii} v^\top x_i x_i^\top v \leq \sum_i v^\top x_i x_i^\top v = v^\top X^\top Xv \leq \lambda_{\max}(X^\top X) v^\top v,$$

implying that $c \leq \lambda_{\max}(X^\top X)$.

Therefore, we have $d \leq \lambda_{\max}(X^\top JX) \leq \lambda_{\max}(X^\top X)$. Then since the eigenvalues of $Z^\top Z$ are the diagonal elements of $D$, the eigenvalues of $D(D^\top D + tI_m)^{-1} D^\top$ are bounded by $\frac{\lambda_{\max}(X^\top X)^2}{\lambda_{\max}(X^\top X)^2 + t}$.

Then recall $t = n\gamma\lambda(1-\alpha)$ and $a^\top a \geq 1$, we have

$$\begin{aligned}
&\mathbf{1}_n^\top \Psi \mathbf{1}_n - \mathbf{1}_n^\top \Psi X_A \left(X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|}\right)^{-1} X_A^\top \Psi \mathbf{1}_n \\
&= \frac{1}{n\gamma}\left(a^\top a - (U^\top a)^\top D(D^\top D + tI_m)^{-1} D^\top (U^\top a)\right) \\
&\geq \frac{1}{n\gamma}(a^\top a - \frac{\lambda_{\max}(X^\top X)^2}{\lambda_{\max}(X^\top X)^2 + t}(U^\top a)^\top U^\top a) \\
&= \frac{1}{n\gamma} \times \frac{n\gamma\lambda(1-\alpha)}{\lambda_{\max}(X^\top X)^2 + n\gamma\lambda(1-\alpha)} a^\top a \\
&\geq \frac{\lambda(1-\alpha)}{\lambda_{\max}(X^\top X)^2 + n\gamma\lambda(1-\alpha)} \\
&> 0.
\end{aligned}$$

Let

$$H_{31} = \begin{bmatrix} \mathbf{1}_n^\top \Psi \mathbf{1}_n & \mathbf{1}_n^\top \Psi X_A \\ X_A^\top \Psi \mathbf{1}_n & X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|} \end{bmatrix},\ H_{32} = \begin{bmatrix} X_B^\top \Psi \mathbf{1}_n & X_B^\top \Psi X_A \end{bmatrix},\ H_{33} = \lambda\alpha I_{|B|}.$$

Observe that $H_{33}^{-1} = \frac{1}{\lambda\alpha} I_{|B|}$. Then if $H_{31}$ is invertible, we have

$$H_3^{-1} = \begin{bmatrix} H_{31}^{-1} & \mathbf{0} \\ -\frac{1}{\lambda\alpha} H_{32} H_{31}^{-1} & \frac{1}{\lambda\alpha} I_{|B|} \end{bmatrix}.$$

Hence to show $H_3$ is invertible, it suffices to show $H_{31}$ is invertible. Let

$$M = X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|},\ b = X_A^\top \Psi \mathbf{1}_n,$$

and

$$\kappa = \mathbf{1}_n^\top \Psi \mathbf{1}_n - \mathbf{1}_n^\top \Psi X_A \left(X_A^\top \Psi X_A + \lambda(1-\alpha) I_{|A|}\right)^{-1} X_A^\top \Psi \mathbf{1}_n.$$

Since $\kappa > 0$, we have

$$H_{31}^{-1} = \begin{bmatrix} \frac{1}{\kappa} & -\frac{1}{\kappa} b^\top M^{-1} \\ -\frac{1}{\kappa} M^{-1} b & M^{-1} + \frac{1}{\kappa} M^{-1} b b^\top M^{-1} \end{bmatrix},$$

and it follows that $H_3$ is invertible.

It can be easily shown that $\|b\| = \|b^\top\| \leq \frac{1}{\sqrt{n\gamma}} \|X\|,\ \|M^{-1}\| \leq \frac{1}{\lambda(1-\alpha)}$. Combine this with $\frac{1}{\kappa} \leq \frac{\lambda_{\max}(X^\top X)^2 + n\gamma\lambda(1-\alpha)}{\lambda(1-\alpha)}$, then similar to (A.6), we have

$$\|H_{31}^{-1}\| \leq \frac{1}{\lambda(1-\alpha)} + \frac{\lambda_{\max}(X^\top X)^2 + n\gamma\lambda(1-\alpha)}{\lambda(1-\alpha)} \left(1 + \frac{\|X\|}{\sqrt{n\gamma\lambda(1-\alpha)}}\right)^2$$

and then

$$\begin{aligned}\|H_3^{-1}\| \;\leq\; & \frac{1}{\lambda\alpha} + \left[\frac{1}{\lambda(1-\alpha)} + \frac{\lambda_{\max}(X^\top X)^2 + n\gamma\lambda(1-\alpha)}{\lambda(1-\alpha)}\left(1 + \frac{\|X\|}{\sqrt{n}\gamma\lambda(1-\alpha)}\right)^2\right] \\ & \left(1 + \frac{2\|X\|}{\sqrt{n}\gamma\lambda\alpha}\right).\end{aligned}$$

□

# APPENDIX B
# PROOFS FOR CHAPTER 2

## B.1 Proof of Lemma 2.4.

*Proof.* (i) By assumption, the Jacobian $J_F$ is continuous at $z$. Since

$$\begin{aligned}
&\frac{\|F(z+h)-F(z)-J_F(z+h)h\|_2}{\|h\|_2} \\
\leq\ & \frac{\|F(z+h)-F(z)-J_F(z)h\|_2+\|(J_F(z)-J_F(z+h))h\|_2}{\|h\|_2} \\
\leq\ & \frac{\|F(z+h)-F(z)-J_F(z)h\|_2}{\|h\|_2}+\|J_F(z)-J_F(z+h)\| \\
\to\ & 0
\end{aligned}$$

as $h \to \mathbf{0}$, by definition $J_F \in \nabla_N F(z)$.

(ii)

$$\|AF(z+h)-AF(z)-AH(z+h)h\|_2 \leq \|A\|\|F(z+h)-F(z)-H(z+h)h\|_2 = o(\|h\|_2),$$

hence $AH \in \nabla_N AF(z)$.

(iii)

$$\begin{aligned}
&\|(F(z+h)+G(z+h))-(F(z)+G(z))-(H_F(z+h)+H_G(z+h))h\|_2 \\
\leq\ & \|F(z+h)-F(z)-H_F(z+h)h\|_2+\|G(z+h)-G(z)-H_G(z+h)h\|_2 \\
=\ & o(\|h\|_2),
\end{aligned}$$

hence $H_F + H_G \in \nabla_N(F+G)(z)$.

(iv) It can be seen by observing that

$$\|F(z+h)-F(z)-H(z+h)h\|_2^2 = \sum_{i=1}^{l}(F_i(z+h)-F_i(z)-H_i(z+h)h)^2.$$

□

## B.2 Proof of Lemma 2.5.

*Proof.* If $f$ is differentiable at $z$ with derivative $f'$ defined in its neighborhood, by smoothness assumption and Lemma 2.4(i), $f' \in \nabla_N f(z)$.

If $f$ is not differentiable at $z$, by assumption there exists $s > 0$ such that $f$ is smooth on both $(z-s, z)$ and $(z, z+s)$ implying that $f'(z-) = \lim_{h\to 0^-} \frac{f(z+h)-f(z)}{h}$ and $f'(z+) = \lim_{h\to 0^+} \frac{f(z+h)-f(z)}{h}$ exist and

$$f'(z+h) \to f'(z-) \qquad \text{as } h \to 0^-,$$
$$f'(z+h) \to f'(z+) \qquad \text{as } h \to 0^+.$$

Hence for any $\varepsilon > 0$, there exists a sufficiently small $\delta > 0$ such that

$$\forall\, x \in (z-\delta, z), \quad \tfrac{|f(x)-f(z)-f'(z-)(x-z)|}{|x-z|} < \varepsilon/2, \quad |f'(x) - f'(z-)| < \varepsilon/2;$$
$$\forall\, x \in (z, z+\delta), \quad \tfrac{|f(x)-f(z)-f'(z+)(x-z)|}{|x-z|} < \varepsilon/2, \quad |f'(x) - f'(z+)| < \varepsilon/2.$$

Thus for $x \in (z-\delta, z)$,

$$\frac{|f(x)-f(z)-f'(x)(x-z)|}{|x-z|} \le \frac{|f(x)-f(z)-f'(z-)(x-z)|}{|x-z|} + |f'(z-) - f'(x)| < \varepsilon,$$

and similarly for $x \in (z, z+\delta)$. Define $H(z)$ as in the lemma, then the above implies

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ s.t. } \forall |x-\delta| < z, \ \frac{|f(x)-f(z)-H(x)(x-z)|}{|x-z|} < \epsilon.$$

In other word, $f$ is Newton differentiable at $z$ with $H \in \nabla_N f(z)$.

□

# APPENDIX C
# PROOFS FOR CHAPTER 3

## C.1 Proof of Theorem 3.4.

*Proof.* Without loss of generality, assume $\theta_k$ has exactly one cluster point $\theta^\star$, i.e. $\theta_k \to \theta^\star$. Notice that

$$|t| - \frac{\gamma}{2} \leq h_\gamma(t) \leq |t|,$$

hence

$$f_A(\theta;\lambda) - \frac{\gamma}{2} \leq f_H(\theta;\lambda,\gamma) \leq f_A(\theta;\lambda).$$

Let $\widehat{\theta}_A$ be a minimizer of $f_A(\theta;\lambda)$, and $f_A^0 = \min_\theta f_A(\theta;\lambda) = f_A(\widehat{\theta}_A;\lambda)$, then

$$f_H(\theta_k;\lambda,\gamma_k) \leq f_H(\widehat{\theta}_A;\lambda,\gamma_k) \leq f_A(\widehat{\theta}_A;\lambda) = f_A^0.$$

For any $\epsilon > 0$, there exists $K$ such that for $k \geq K$, $\gamma_k < 2\epsilon$, then

$$f_H(\theta_k;\lambda,\gamma_k) \geq f_A(\theta_k;\lambda) - \epsilon \geq f_A^0 - \epsilon.$$

Hence for $k \geq K$,

$$f_A^0 - \epsilon \leq f_A(\theta_k;\lambda) - \epsilon \leq f_A^0.$$

Let $k \to \infty$, we have $f_A^0 \leq f_A(\theta^\star) \leq f_A^0 + \epsilon$. Since $\epsilon$ is arbitrary, $f_A(\theta^\star) = f_A^0$. □

## C.2 Proof of Theorem 3.5.

*Proof.* Without loss of generality, assume $\theta_k$ has exactly one cluster point $\theta^\star$, i.e. $\theta_k \to \theta^\star$. Notice that

$$\gamma h_\gamma(t) \leq \frac{1}{2}t^2,$$

which implies

$$\gamma f_H(\theta; \lambda/\gamma, \gamma) \leq f_S(\theta; \lambda).$$

Let $\widehat{\theta}_S$ be a minimizer of $f_S(\theta; \lambda)$, and $f_S^0 = \min_\theta f_S(\theta; \lambda) = f_S(\widehat{\theta}_S; \lambda)$, then

$$\gamma_k f_H(\theta_k; \lambda/\gamma_k, \gamma_k) \leq \gamma_k f_H(\widehat{\theta}_S; \lambda/\gamma_k, \gamma_k) \leq f_S(\widehat{\theta}_S; \lambda) = f_S^0.$$

Since $\theta_k = (\beta_0^k, \beta^k)$ is convergent, $r^k = y - \beta_0^k \mathbf{1} - X\beta^k$ is convergent too. Then there exists $M > 0$ such that $\|r^k\|_\infty \leq M$. There exists $K$ such that for $k \geq K$, $\gamma_k > M$, then $h_\gamma(r_i k) = \frac{1}{2} r_i k^2$, and

$$\gamma_k f_H(\theta_k; \lambda/\gamma_k, \gamma_k) = f_S(\theta_k; \lambda).$$

Hence for $k \geq K$,

$$f_S(\theta_k; \lambda) \leq f_S^0.$$

Let $k \to \infty$, we have $f_S(\theta^\star; \lambda) \leq f_S^0$. Since $f_S(\theta^\star; \lambda) \geq \min_\theta f_S(\theta; \lambda) = f_S^0$, $f_S(\theta^\star) = f_S^0$. □

# APPENDIX D
# AUXILIARY RESULTS FOR CHAPTER 4

**Lemma D.1.** *Under the assumptions of Corollary 4.3, for any $v \in \mathbb{R}^n$, $w \in \mathbb{R}^p$ we have*

$$P(|\frac{1}{n}v^\top DXw| > c_0\sigma_d\sigma_x\frac{\log(n+p)}{n}\|v\|_1\|w\|_2) \le c_1\exp(-c_2\log(n+p)),$$

*where $c_0, c_1, c_2 > 0$.*

*Proof.* Note that

$$|\frac{1}{n}v^\top DXw| = \frac{1}{n}|\sum_{i=1}^n v_i d_i x_i^\top w| \le \frac{1}{n}\sum_{i=1}^n |v_i d_i| \max_{1\le i\le n}|x_i^\top w|,$$

hence for $t_1, t_2 > 0$,

$$\begin{aligned} P(|\frac{1}{n}v^\top DXw| > t_1t_2) &\le P(\sum_{i=1}^n |v_i d_i| \max_{1\le i\le n}|x_i^\top w| \ge nt_1t_2) \\ &\le P(\sum_{i=1}^n |v_i d_i| \ge \sqrt{n}t_1) + P(\max_{1\le i\le n}|x_i^\top w| \ge \sqrt{n}t_2) \end{aligned}$$

Note that the sub-Gaussian norm

$$\|\sum_{i=1}^n |v_i d_i|\|_{\psi_2} \le \sum_{i=1}^n |v_i|\|d_i\|_{\psi_2} \le \sum_{i=1}^n |v_i|\sigma_d = \|v\|_1\sigma_d < \infty,$$

which implies that $\sum_{i=1}^n |v_i d_i|$ is sub-Gaussian with parameter at most $\|v\|_1\sigma_d$. By the tail decay property, there exists $c > 0$ such that

$$P(\sum_{i=1}^n |v_i d_i| > t) \le \exp(1 - ct^2/\|\sum_{i=1}^n |v_i d_i|\|_{\psi_2})^2, \quad \forall t \ge 0.$$

Then using $t_1 = \sigma_d\sqrt{\frac{\log(n+p)}{n}}\|v\|_1$ yields

$$P\left(\sum_{i=1}^n |v_i d_i| > \sqrt{n}\cdot\sigma_d\sqrt{\frac{\log(n+p)}{n}}\|v\|_1\right) \le \exp(1 - c\log(n+p)).$$

Similarly, $x_i^\top w/\|w\|_2$ is sub-Gaussian with parameter $\sigma_x$ for all $i$ by assumption,

implying the existence of $c' > 0$ such that

$$P(|x_i^\top w|/\|w\|_2 > t) \leq \exp(1 - c't^2/\sigma_x^2), \quad \forall 1 \leq i \leq n, \forall t \geq 0.$$

Let $t_2 = c_0\sigma_x\sqrt{\frac{\log(n+p)}{n}}\|w\|_2$, where $c_0 > \frac{1}{c'}$, then we have

$$\begin{aligned} P(\max_{1\leq i\leq n} |x_i^\top w| > \sqrt{n}\cdot c_0\sigma_x\sqrt{\frac{\log(n+p)}{n}}\|w\|_2) &\leq \sum_{i=1}^n P(\frac{|x_i^\top w|}{\|w\|_2} \geq c_0\sigma_x\sqrt{\log(n+p)}) \\ &\leq n\exp(1 - c'c_0\log(n+p)) \\ &\leq \exp(1 - (c'c_0 - 1)\log(n+p)). \end{aligned}$$

Finally, combining the results above yields

$$P(|\frac{1}{n}v^\top DXw| > c_0\sigma_d\sigma_x\frac{\log(n+p)}{n}\|v\|_1\|w\|_2) \leq c_1\exp(-c_2\log(n+p)),$$

for some $c_1, c_2 > 0$. □

**Lemma D.2.** *Under the assumptions of Corollary 4.3, we have*

$$P\left(\|\nabla\mathcal{L}_n(\theta^*)\|_\infty > c_0\max(\sigma_d,\sigma_x)\sigma_\varepsilon\sqrt{\frac{\log(n+p)}{n}}\right) \leq c_1\exp(-c_2\log(n+p)).$$

*for some* $c_0, c_1, c_2 > 0$.

*Proof.*

$$\begin{aligned} \|\nabla\mathcal{L}_n(\theta^*)\|_\infty &= \|\frac{1}{n}(D\ X)^\top((D\ X)\theta^* - y)\|_\infty \\ &= \|\frac{1}{n}(D\ X)^\top\varepsilon\|_\infty \\ &= \|\frac{1}{n}D\varepsilon\|_\infty + \|\frac{1}{n}X^\top\varepsilon\|_\infty. \end{aligned}$$

For the second term, Lemma 14 of Loh and Wainwright (2012) implies

$$P(\|\frac{1}{n}X^\top\varepsilon\|_\infty > t) \leq 6p\exp(-cn\min(\frac{t^2}{(\sigma_x\sigma_\varepsilon)^2}, \frac{t}{\sigma_x\sigma_\varepsilon})).$$

For the first term,

$$\begin{aligned} P(\|\frac{1}{n}D\varepsilon\|_\infty > t) &= P(\max_{1\le i\le n}|d_i\varepsilon_i| > nt) \\ &\le \sum_{i=1}^{n} P(|d_i\varepsilon_i| > nt). \end{aligned}$$

Since $d_i$, $\varepsilon_i$ are sub-Gaussian with parameter $\sigma_d$, $\sigma_\varepsilon$ respectively, the rescaled variables $\widetilde{d}_i = d_i/\sigma_d$, $\widetilde{\varepsilon}_i = \varepsilon_i/\sigma_\varepsilon$ are sub-Gaussian with parameter 1. In addition, $\widetilde{d}_i + \widetilde{\varepsilon}_i$ are sub-Gaussian with parameter at most 2. Hence $\widetilde{d}_i^2$, $\widetilde{\varepsilon}_i^2$ and $(\widetilde{d}_i + \widetilde{\varepsilon}_i)^2$ are sub-exponential. Since $E\widetilde{d}_i\widetilde{\varepsilon}_i = 0$, denote $s(v) = v^2 - Ev^2$, $\widetilde{t} = t/(\sigma_d\sigma_\varepsilon)$, then

$$\widetilde{d}_i\widetilde{\varepsilon}_i = \frac{1}{2}\left[s(\widetilde{d}_i + \widetilde{\varepsilon}_i) - s(\widetilde{d}_i) - s(\widetilde{\varepsilon}_i)\right],$$

and

$$\begin{aligned} P(|d_i\varepsilon_i| > nt) &= P(|\widetilde{d}_i\widetilde{\varepsilon}_i| > n\widetilde{t}) \\ &\le P(|s(\widetilde{d}_i + \widetilde{\varepsilon}_i)| > n\widetilde{t}) + P(|s(\widetilde{d}_i)| > \frac{n\widetilde{t}}{2}) + P(|s(\widetilde{\varepsilon}_i)| > \frac{n\widetilde{t}}{2}). \end{aligned}$$

Thus we may apply a Bernstein-type inequality (Proposition 5.16 of Vershynin (2010)) to each term above and obtain

$$P(\|\frac{1}{n}D\varepsilon\|_\infty > t) \le nP(|d_1\varepsilon_1| > nt) \le 6n\exp(-c'n\min(\frac{t^2}{(\sigma_d\sigma_\varepsilon)^2}, \frac{t}{\sigma_d\sigma_\varepsilon})).$$

Finally, set $t = c_0\max(\sigma_d, \sigma_x)\sigma_\varepsilon\sqrt{\frac{\log(n+p)}{n}}$, where $c_0 > \frac{1}{\sqrt{\min(c,c')}}$. Then with the assumption $n \gtrsim \log(n+p)$ we obtain the inequality

$$P\left(\|\nabla\mathcal{L}_n(\theta^*)\|_\infty > c_0\max(\sigma_d, \sigma_x)\sigma_\varepsilon\sqrt{\frac{\log(n+p)}{n}}\right) \le c_1\exp(-c_2\log(n+p)).$$

□

## REFERENCES


Agarwal, A., S. Negahban, and M. Wainwright (2012). Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics 40*(5), 2452–2482.

Breheny, P. and J. Huang (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics 5*(1), 232–253.

Bühlmann, P., M. Kalisch, and L. Meier (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application 1*, 255–278.

Chen, X., Z. Nashed, and L. Qi (2000). Smoothing methods and semismooth methods for nondifferentiable operator equations. *SIAM Journal on Numerical Analysis 38*(4), 1200–1216.

Clarke, F. H. (1983). *Optimization and Nonsmooth Analysis*. Wiley.

Clarke, F. H. (1990). *Optimization and nonsmooth analysis*, Volume 5. Siam.

Combettes, P. L. and V. R. Wajs (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation 4*(4), 1168–1200.

Everitt, B. S. (1981). *Finite mixture distributions*. Wiley Online Library.

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

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics 1*(2), 302–332.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

Hampel, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, 1887–1896.

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

Holland, P. W. and R. E. Welsch (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods 6*(9), 813–827.

Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics 1*(5), 799–821.

Huber, P. J. (1981). Robust statistics.

Ito, K. and K. Kunisch (2008). *Lagrange Multiplier Approach to Variational Problems and Applications*. Philadelphia, PA: SIAM.

Koenker, R. and G. Bassett Jr (1978). Regression quantiles. *Econometrica 46*(1), 33–50.

Koenker, R. and J. A. Machado (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association 94*(448), 1296–1310.

Li, Y. and G. R. Arce (2004). A maximum likelihood approach to least absolute deviation regression. *EURASIP Journal on Advances in Signal Processing 2004*(12), 1–8.

Loh, P.-L. and M. J. Wainwright (2012). High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *The Annals of Statistics 40*(3), 1637–1664.

Loh, P.-L. and M. J. Wainwright (2015). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *The Journal of Machine Learning Research 16*(1), 559–616.

Mifflin, R. (1977). Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization 15*(6), 959–972.

Osborne, M. and B. Turlach (2011). A homotopy algorithm for the quantile regression lasso and related piecewise linear problems. *Journal of Computational and Graphical Statistics 20*(4), 972–987.

Peng, B. and L. Wang (2015). An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression. *Journal of Computational and Graphical Statistics 24*(3), 676–694.

Portnoy, S., R. Koenker, et al. (1997). The gaussian hare and the laplacian tortoise: Computability of squared-error versus absolute-error estimators. *Statistical Science 12*(4), 279–300.

Qi, L. and J. Sun (1993). A nonsmooth version of newton's method. *Mathematical Programming 58*(1–3), 353–367.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton, NJ: Princeton University Press.

Rousseeuw, P. and V. Yohai (1984). Robust regression by means of s-estimators. In *Robust and nonlinear time series analysis*, pp. 256–272. Springer.

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

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

She, Y. and A. B. Owen (2012). Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2011). Regularization paths for coxs proportional hazards model via coordinate descent. *Journal of Statistical Software 39*(5), 1–13.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 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. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74*(2), 245–266.

Tsanas, A. and A. Xifara (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings 49*, 560–567.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications 109*(3), 475–494.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Wang, H., B. Li, and C. Leng (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(3), 671–683.

Wang, H., R. Li, and C.-L. Tsai (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika 94*(3), 553–568.

Wang, L., Y. Wu, and R. Li (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association 107*(497), 214–222.

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

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

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics 38*(2), 894–942.

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