Rank Based Group Variable Selection for Functional Linear Model

by

Jieun Park

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 14, 2019

Keywords: Robust, Rank-Based, Variable Selection, Functional Analysis

Approved by

Asheber Abebe, Chair, Professor of Mathematics and Statistics
Nedret Billor, Co-chair, Professor of Mathematics and Statistics
Guanqun Cao, Associate Professor of Mathematics and Statistics
Peng Zeng, Professor of Mathematics and Statistics
Jay Khodadadi, Professor of Mechanical Engineering

Abstract

We propose a robust rank based variable selection method for a functional linear regression model with multiple explanatory functions and a scalar response. The procedure extends rank based group variable selection to functional variable selection and the proposed estimator is robust in the presence of outliers in predictor function space as well as response space. The performance of the proposed robust method is demonstrated with an extensive simulation study and real data examples. We prove the proposed method with a group-adaptive penalty achieves the oracle property.

Acknowledgments

I would like to express my deepest appreciation to my advisors, Drs. Asheber Abebe and Nedret Billor for their support and guidance. It would not have been possible to complete this dissertation without them during the particularly difficult moments. They have provided me with guidance and patience when I struggled and pushed me toward a successful path. I learned tremendous and valuable lesson from them and am fortunate to have had the opportunity to work with them.

I would also like to thank my committee members Drs. Guanqun Cao, Peng Zeng and Jay Khodadadi for taking time to read over my dissertation and provide helpful advice. Their comments have been helpful in shaping and framing my research.

This dissertation was also made possible by the support and help of my friends at the Department of Mathematics and Statistics, my friends outside it as well.

I would not be where I am today without unwavering support of my family. My sisters have all been supportive when support was needed. Most of all thanks to my husband, Youngsoo, and my children, Eleanor and Elliot, for their support and love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the important topics in statistics is the study of the relationship among variables via regression models. Linear regression analysis, in particular, is fundamental for the functional data analysis which is the analysis of infinite-dimensional variables as curves, images, and time-variant inputs.

Throughout this dissertation, we consider a functional multiple linear regression model with p functional predictors and a continuous scalar response defined by

$$y_i = \alpha + \sum_{j=1}^{p} \int_{\mathcal{T}} x_{ij}(t)\beta_j(t)dt + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1.1}$$

where y_i is a scalar and $x_{ij}(t)$'s on $\mathcal{T}$, the support of functional covariates, are L^2 integrable and independent with each other, $\beta_j(t)$'s are functional parameters which are also L^2 integrable, and $\varepsilon_i \overset{iid}{\sim} F$, where F is a distribution with finite Fisher information. We follow the same definition and notation by Gertheiss et al [12].

When a basis expansion with d basis is used for functional predictors and functional parameters, there are $d \times p$ predictors in the multiple linear regression model. Thus identifying the subset of significant predictor functions becomes a group variable selection problem rather than a single variable level selection problem.

There are various variable selection methods. Recently, some elaborated techniques have been devised compared to the traditional methods such as forward, backward, and stepwise selections based on C_p, AIC, or BIC. As one of the modern analyses, the regularized estimation methods were applied to select meaningful variables since Tibshirani [41] proposed the Lasso.

The Lasso minimizes the residual sum of squares under the penalty with the ℓ_1 norm of the coefficients. It estimates parameters and selects a sparse subset simultaneously. Fan and Li [10] suggested the smoothly clipped absolute deviation (SCAD) with less biases than Lasso. Zou [47] proposed the adaptive Lasso with a different amount of contribution of each coefficient depending on their sizes. Fan and Li [10] and Zou [47] showed that their methods have oracle properties, that is, their estimators have efficiency and consistency with proper choice of regularization parameters.

More recently, like our problems, selecting significant groups of variables has received significant attention. Yuan and Lin [46] extended the Lasso to the group Lasso which selects groups of variables under the penalty with the sum of the weighted ℓ_2 norms of the coefficients on each group of variables. Wang and Leng [42] obtained the consistency and the oracle property of their estimator by proposing the adaptive group Lasso. By using these least square based group variable selection methods, Gertheiss et al. [12] proposed a variable selection method for multiple functional linear regression models. The method minimizes the residual sum of squares loss function under the penalty which controls both smoothness and sparsity. The ℓ_1 penalty plays an important role in making a parameter exactly zero, that is, removing non-significant variables.

The aforementioned techniques are based on ℓ_2 norm loss minimization. They are efficient if the true underlying distribution follows the normal distribution. However, the ℓ_2 type of objective functions is vulnerable when the data contain outliers or are heavy-tailed. The goal of this dissertation is to develop another version of the group Lasso technique that is applied to a functional linear model with the shortcoming of ℓ_2 loss minimization methods removed.

Several trials have been made to overcome those drawbacks and to achieve robustness in multiple linear regression models. Rosset and Zhu [39] proposed Huberized Lasso with a loss function similar to Huber's loss function. Owen [36] also proposed a robust hybrid of the Lasso and the ridge regression. Some researchers used the quantile regression loss function called the check function in Koenker and Bassett [27]. Koenker [25] optimized the Lasso regularized quantile regression with the original ℓ_1 Lasso penalty. Wang et al. [43] obtained robust estimators by combining the least absolute deviation (LAD) regression and the Lasso,

that is, ℓ_1 norm loss for the residuals and the ℓ_1 norm penalty for coefficients. Their work is a special case of Koenker's [25] model which has $\tau = 0.5$ in quantile regression. To achieve the oracle property of the parameter estimators, Wu and Liu [45] suggested the adaptive Lasso regularized quantile regression with the adaptive Lasso penalty.

Johnson and Peng [20] suggested a robust variable selection approach by replacing the ℓ_2 loss function with Jaeckel's [18] dispersion function and the same Lasso ℓ_1 norm penalty as in the original Lasso. Abebe and Bindele [1] proposed a robust variable selection procedure based on a weighted signed-rank loss function with Lasso penalty. Wang and Li [44] used the same rank based loss function as Johnson and Peng [20], and with the weight on the leverage of predictors and the smoothly clipped absolute deviation (SCAD) penalty for the coefficients. Thus Wang and Li [44] achieved robustness in both the predictor space and the response space, whereas Johnson and Peng [20] obtained robustness in the response space only. Bindele, Abebe and Zeng [5] used the same weighted signed-rank loss function with Lass penalty for variable selection in single-index models.

Some regularization methods have been developed to select grouped variables robustly. Group variable selection method based on quantile regression proposed by Kato [22] and Bang and Jhun [4]. Kato [22] employed group l_2 penalty (the original group Lasso penalty in Yuan and Lin [46]) with the quantile regression. Bang and Jhun [4] proposed the adaptive sup-norm regularized quantile regression which penalizes the check loss function [27] by the sum of group-wise adaptive sup-norm penalties. It is proven that the method satisfies the oracle property. Miakonkana et al. [32] penalized a weighted rank-based loss function same as the one in Wang and Li [44], with a group adaptive Lasso ℓ_1 norm penalty function. It is shown that the weighted rank-based group adaptive Lasso method achieves robustness in both the response space and the predictor space as well as selecting meaningful group variables.

A traditional functional regression stems from the least square minimization technique. Researchers including Ramsay and Silverman [38] expressed a functional linear model as an ordinary multiple linear regression model using a linear basis expansion on an infinite dimensional functional space. Escabias et al. [9] uses the principal component analysis method to fit a functional logistic regression model. James [19] applied functional principal components to a

general functional linear model and showed that it is useful when only fragments of each curve have been observed. Müller and StadtMüller [34] also discuss the generalized functional linear model, consider estimation methods for its parameters, and select variables based on AIC. However, the estimated parameters by these methods are greatly affected by outliers. Thus some approaches are proposed to achieve robustness in functional regression model. Bali et al. [3] use a robust scale function with functional principal components technique. Sawant et al. [40] discuss robust functional principal components for a functional response and a predictor. Denhere and Billor [7] showed that their method based on the functional principal component analysis eliminates multicollinearity and reduces the effect of functional outliers for a logistic functional linear model. Denhere and Bindele [8] proposed rank estimation for functional linear regression model with a scalar response and functional predictors.

Researches wanted to select and estimate meaningful variables simultaneously. Most variable selection techniques for a functional regression model were based on the regularization method. The penalization shrinks coefficient functions depending on tuning parameters and eventually selects the significant variables. We note that a variable selection problem of functional linear regression model can be understood as a group variable selection of multiple linear regression models. In this point of view, Matsui and Konishi [30] used group SCAD penalty to select variables for a functional linear model with a scalar response and functional predictors. Mingotti et al. [33] proposed "Functional Lasso" for a functional response with scalar predictors by adapting Lasso method to functional linear model. Hone and Lian [15] applied the Lasso regularization method for a functional response with functional predictors to solve a linear ordinary differential equation. However, we consider the functional aspects of coefficient function $\beta(t)$ rather than applying methods for group variable selection directly. Gertheiss et al. [12] included the functional smoothness condition of coefficient functions in the penalty term while penalizing the sum of ℓ_2 norm of the coefficient functions for the generalized linear functional regression model. The ℓ_2 loss function has the same drawback under the existence of outliers even though the ℓ_2 penalty selects functional variables. To overcome this, some robust loss approach has been proposed. Pannu and Billor [37] applied the least absolution deviation method to functional linear model using Gertheiss' penalty function and showed a

robustness of their method. Also, we consider the smoothness property of functions to define a regularization method with a robust loss functions.

In method that depend on regularization, the most difficult part is finding the optimal tuning parameter which minimizes the objective function under the regularization. Researchers have used BIC (Bayesian information criterion), SIC (Schwarz information criterion), AIC (Akaike information criterion), GACV (generalized approximate cross-validation), and GCV (generalized cross-validation) including the traditional CV (cross-validation). We will use SIC in [4] and CV to estimate coefficient functions.

To this end, we propose a robust variable selection method for a functional linear regression model. The rank-based functional regression model is developed by modifying the work of Miakonkana et al. [32] with the penalty function in Gertheiss et al. [12]. Since the model has a weighted rank-based loss function, it has robustness in both the predictor space and the response space. The proposed model conserves the smoothness of coefficient functions while selecting significant functional variables. Also, the adaptive penalty term implies the oracle property. In Chapter 3, we review various versions of regularized group variable selection methods for discrete multiple linear regression. Furthermore, we compare different kinds of loss functions and penalty functions by illustrating their geometric properties. Chapter 4 builds up the notations for the functional regression model. Section 4.1 presents the objective function of Gertheiss et al. [12]. We introduce the proposed method and show its performance compared to the Gertheiss et al.[12]'s work by simulation studies in Chapter 5. We prove the oracle property of estimators as well. An application to Japanese weather data is presented in Section 5.5.

Chapter 2

Multiple Linear Regression

Let y be a response variable and $\mathbf{x}$ be a column vector of p predictors. Suppose that both $\mathbf{x}$ and y are random and that we have their random sample $(\mathbf{x}_i^T, y_i)$, $i = 1, \ldots, n$. The multiple linear regression model is defined as

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad \text{for } i = 1, \ldots, n \tag{2.1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is unknown and ε has $E(\varepsilon_i) = 0$.

2.1 Least Squares and Least Absolute Deviations Regression

The method of least squares (LS) and the method of least absolute deviations (LAD) have been widely used to estimate $\boldsymbol{\beta}$ in Equation (2.1).

The LS method finds the estimated $\boldsymbol{\beta}$ by minimizing the following objective function

$$\widehat{\boldsymbol{\beta}}_{LS} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 \tag{2.2}$$

It was discovered independently by Carl F. Gauss around 1795 and by Adrian M. LeGendre in France around 1805 in [6]. The LS method performs well when the noise ε_i follows a normal distribution. The LAD method works better to deal with the case in which the response has outliers or the noise follows a heavy-tailed distribution. It does not depend on the assumption of normality. The LAD method was presented about 50 years before the LS method, in 1757 by Roger Joseph Boscovich [6]. The LS method overwhelmed the LAD for several decades due

to the relative ease and simplicity of the LS method. However, nowadays, computation is not a hurdle and its theory of tests for parameters has been established [26, 2]. The LAD method estimates $\boldsymbol{\beta}$ by minimizing

$$\widehat{\boldsymbol{\beta}}_{LAD} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} |y_i - \mathbf{x}_i^T\boldsymbol{\beta}|. \tag{2.3}$$

The LAD method is optimal when ε_i follows the Laplace distribution. Moreover, LAD can be understood as a special case of Quantile regression when $\tau = 0.5$ in the following equation in Koenker et al. [28].

$$\widehat{\boldsymbol{\beta}}_{Quantile} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^T\boldsymbol{\beta}) \tag{2.4}$$

where $\rho_\tau(u) := \{\tau - I(u \le 0)\} \cdot u$ is the check loss function [27]. For $\tau = 0.5$, Equation (2.4) will be the following equation and $\widehat{\boldsymbol{\beta}}_{Quatile,\tau=0.5} = \widehat{\boldsymbol{\beta}}_{LAD}$.

$$\widehat{\boldsymbol{\beta}}_{Quatile,\tau=0.5} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}|y_i - \mathbf{x}_i^T\boldsymbol{\beta}| \tag{2.5}$$

since the check loss function becomes $\rho_{0.5}(u) = \frac{1}{2}|u|$. Apparently, $\widehat{\boldsymbol{\beta}}_{Quatile,\tau=0.5} = \widehat{\boldsymbol{\beta}}_{LAD}$. Henceforward, for LAD method, we refer to the result of Quantile regression with $\tau = 1/2$.

2.2 Rank-Based Regression

The goal of the rank-based regression method is to estimate the coefficient vector $\boldsymbol{\beta}$ in Equation (2.1). The rank-based method pursues also to estimate the parameter $\boldsymbol{\beta}$ under the presence of outliers similar to LAD. We assume that the errors are independent and identically distributed (iid) with a continuous probability density function (pdf) $f(t)$. Let $\mathbf{y} = (y_1, \ldots, y_n)^T$ be the $n \times 1$ vector of responses, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ the $n \times p$ design matrix, and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T$ the $n \times 1$ error vector. Then we can rewrite Equation (2.1) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon. \tag{2.6}$$

LS regression minimizes the Euclidean distance and LAD regression minimizes the absolute deviations measure between $\mathbf{y}$ and $\mathbf{X}\widehat{\boldsymbol{\beta}}$ which is the estimated $\mathbf{y}$. We define a new distance measure to achieve the rank-based estimator for the coefficient vector $\boldsymbol{\beta}$ based on Jaeckel's dispersion function [18]. We follow the notations and terminology by Jaeckel [18] and Jurečková [21].

Before defining the rank-based method, we introduce the definition of a *pseudo-norm* as in Hettmansperger and McKean [14]. An operator $||\cdot||_\varphi$ is called a *pseudo-norm* if it satisfies the following four conditions.

1. $||\mathbf{u}+\mathbf{v}||_\varphi \leq ||\mathbf{u}||_\varphi + ||\mathbf{v}||_\varphi$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$
2. $||\alpha\mathbf{u}||_\varphi = |\alpha|||\mathbf{u}||_\varphi$ for all $\alpha \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^n$
3. $||\mathbf{u}||_\varphi \geq 0$ for all $\mathbf{u} \in \mathbb{R}^n$
4. $||\mathbf{u}||_\varphi = 0$ if and only if $u_1 = \cdots = u_n$

Jaeckel's dispersion function measuring the distance between two vectors is defined by

$$D(\boldsymbol{\beta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_\varphi, \tag{2.7}$$

where

$$||\mathbf{u}||_\varphi = \sum_{i=1}^{n} a(R(u_i))u_i, \tag{2.8}$$

R denotes the rank, $a(t) = \varphi(\frac{t}{n+1})$, and φ is a nondecreasing and L^2-integrable score function defined on the interval $[0,1]$ as in Kloke and McKean [24]. Without loss of generality, we assume $\int \varphi(s)ds = 0$ and $\int \varphi^2(s)ds = 1$. Then one can check $||\cdot||_\varphi$ in Equation (2.8) is a pseudo-norm. A primal-dual relationship between quantile regression and rank estimation is given in Gutenbrunner and Jurecková [13].

Let φ be Wilcoxon score, that is, $\varphi\Big(\frac{t}{n+1}\Big) = \frac{t}{n+1} - \frac{1}{2}$. Then, Jaeckel's Wilcoxon-type dispersion function $D(\boldsymbol{\beta})$ can be written as

$$D(\boldsymbol{\beta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_{\varphi} \tag{2.9}$$

$$= \frac{1}{2(n+1)} \sum_{i<j} |\varepsilon_i - \varepsilon_j| \tag{2.10}$$

Johnson and Peng [20] used the following objective function similar to Equation (2.10) for the linear regression model in Equation (2.1).

$$\sum_{i<j} |\varepsilon_i - \varepsilon_j| \tag{2.11}$$

Furthermore, to achieve robustness in the predictor space, Wang and Li [44] proposed the weighted rank-based loss function

$$\sum_{i<j} b_{ij} |\varepsilon_i - \varepsilon_j| \tag{2.12}$$

where

$$b_{ij} = b(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i) h(\mathbf{x}_j), \tag{2.13}$$

which degrades high leverage points, where

$$h(\mathbf{x}_i) = \min \left[1, \frac{b}{(\mathbf{x}_i - \hat{\mu})^T S^{-1} (\mathbf{x}_i - \hat{\mu})} \right] \tag{2.14}$$

with $(\hat{\mu}, S)$ being the robust minimum volume ellipsoid estimators of the location and spread as in Wang and Li [44] and Miakonkana et al. [32].

We call this weighted Wilcoxon-type rank-based method as the *rank-based* regression method. The rank-based (RB) method estimates β by minimizing the following weighted Wilcoxon-type dispersion function.

$$\widehat{\boldsymbol{\beta}}_{RB} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i<j} b_{ij} |\varepsilon_i - \varepsilon_j| \tag{2.15}$$

where b_{ij} is defined by Equation (2.13). We use the rank-based method as the loss function for the proposed rank-based penalized method for functional linear regression model.

2.3 Loss Functions

We define three loss functions.

1. The L_2 loss is written as the square of ℓ_2 norm of residuals

$$Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 \tag{2.16}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2. \tag{2.17}$$

2. The L_1 loss is the ℓ_1 norm of residuals

$$Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}|y_i - \mathbf{x}_i^T\boldsymbol{\beta}| \tag{2.18}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}|\varepsilon_i|. \tag{2.19}$$

3. The rank-based loss function L_{RB} [32, 20, 44] is

$$Loss_{RB}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i<j} b_{ij}|(y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (y_j - \mathbf{x}_j^T\boldsymbol{\beta})| \tag{2.20}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|\varepsilon_i - \varepsilon_j|. \tag{2.21}$$

The kinds of loss functions determine the robustness of estimation under the existence of outliers when we solve a linear regression problem (2.1).

2.4 Penalty Function for the Regularized Method

The variable selection is important for high dimensional models. Traditional approaches such as forward, backward, and stepwise selections are computationally expensive and unstable. Alternative approach has been made under the sparsity assumption. We assume that the true

model has contributions of a few meaningful covariates and that other nonsignificant covariates do not affect the model completely.

For the regularized method, several penalty functions have been proposed. The original Lasso is proposed by Tibshirani [41]. For the discrete original multiple linear regression, we will consider only the original Lasso penalty, that is, ℓ_1 penalty for $\boldsymbol{\beta}$. The discrete non-grouped ℓ_1 penalty is

$$p_{\lambda,\ell_1}(\boldsymbol{\beta}) = \frac{\lambda}{n}||\boldsymbol{\beta}||_1 = \frac{\lambda}{n}\sum_{j=1}^{p}|\beta_j|. \tag{2.22}$$

The component adaptive Lasso penalty function is

$$p_{\lambda,a\ell_1}(\boldsymbol{\beta}) = \frac{\lambda}{n}||\boldsymbol{\lambda}_a\boldsymbol{\beta}||_1 = \frac{\lambda}{n}\sum_{j=1}^{p}|\boldsymbol{\lambda}_{aj}\beta_j|, \tag{2.23}$$

where $\boldsymbol{\lambda}_{aj} = 1/|\tilde{\beta}_j|$ for a suitable initial estimate $\tilde{\beta}_j$ of β_j, $j = 1, \ldots, p$.

2.5 Objective Functions

We consider the objective functions for this dissertation by combining LS, LAD, and RB loss functions with the ℓ_1 penalty function.

2.5.1 LS-Based Objective Functions

LS_ℓ_1 method is the traditional Lasso by Tibshirani [41]. The objective function is combined by the loss function L_{LS} and the ℓ_1 penalty.

$$Q_{LS\_\ell_1}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,\ell_1}(\boldsymbol{\beta}) \tag{2.24}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \frac{\lambda}{n}||\boldsymbol{\beta}||_1 \tag{2.25}$$

$LS\_a\ell_1$ method is the adaptive Lasso. The objective function is combined by the loss function L_{LS} and the adaptive ℓ_1 penalty $p_{\lambda,a\ell_1}(\boldsymbol{\beta})$.

$$Q_{LS\_a\ell_1}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,a\ell_1}(\boldsymbol{\beta}) \tag{2.26}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \frac{\lambda}{n}\sum_{j=1}^{p}\frac{|\beta_j|}{|\tilde{\beta}_j|} \tag{2.27}$$

where $\tilde{\beta}_j$ is an initial estimate of β_j, $j = 1, \ldots, p$.

2.5.2 LAD-Based Objective Functions

The objective function is combined by the loss function L_{LAD} and the ℓ_1 penalty. We use the quantile regression loss function with $\tau = 1/2$ for LAD regression.

$$Q_{LAD\_\ell_1}(\boldsymbol{\beta}) = Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,\ell_1}(\boldsymbol{\beta}) \tag{2.28}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}|y_i - \mathbf{x}_i^T\boldsymbol{\beta}| + \frac{\lambda}{n}||\boldsymbol{\beta}||_1 \tag{2.29}$$

$LAD\_a\ell_1$ method is the adaptive Lasso. The objective function is combined by the loss function L_{LAD} and the adaptive ℓ_1 penalty $p_{\lambda,a\ell_1}(\boldsymbol{\beta})$.

$$Q_{LAD\_a\ell_1}(\boldsymbol{\beta}) = Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,a\ell_1}(\boldsymbol{\beta}) \tag{2.30}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}|y_i - \mathbf{x}_i^T\boldsymbol{\beta}| + \frac{\lambda}{n}\sum_{j=1}^{p}\frac{|\beta_j|}{|\tilde{\beta}_j|} \tag{2.31}$$

where $\tilde{\beta}_j$ is an initial estimate of β_j, $j = 1, \ldots, p$.

2.5.3 RB-Based Objective Functions

The objective function is combined by the loss function L_{RB} and the ℓ_1 penalty. We use the quantile regression loss function with $\tau = 1/2$ for RB regression.

$$Q_{RB\_\ell_1}(\boldsymbol{\beta}) = Loss_{RB}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,\ell_1}(\boldsymbol{\beta}) \tag{2.32}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|(y_i - \mathbf{x}_i^T\beta) - (y_j - \mathbf{x}_j^T\beta)| + \frac{\lambda}{n}||\boldsymbol{\beta}||_1 \tag{2.33}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|\varepsilon_i - \varepsilon_j| + \frac{\lambda}{n}||\boldsymbol{\beta}||_1 \tag{2.34}$$

$RB\_a\ell_1$ method is the adaptive Lasso. The objective function is combined by the loss function L_{RB} and the adaptive ℓ_1 penalty $p_{\lambda,a\ell_1}(\boldsymbol{\beta})$.

$$Q_{RB\_a\ell_1}(\boldsymbol{\beta}) = Loss_{RB}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + p_{\lambda,a\ell_1}(\boldsymbol{\beta}) \tag{2.35}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|(y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (y_j - \mathbf{x}_j^T\boldsymbol{\beta})| + \frac{\lambda}{n}\sum_{j=1}^{p}\frac{|\beta_j|}{|\tilde{\beta}_j|} \tag{2.36}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|\varepsilon_i - \varepsilon_j| + \frac{\lambda}{n}\sum_{j=1}^{p}\frac{|\beta_j|}{|\tilde{\beta}_j|} \tag{2.37}$$

where $\tilde{\beta}_j$ is an initial estimate of β_j, $j = 1, \ldots, p$.

Chapter 3

Grouped Multiple Linear Regression

3.1 Grouped Multiple Linear Model

We summarize the existing regularized estimation method for group variable selection in multiple linear regression model. We consider the following multiple linear regression model with K groups as follows.

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{X}_k \boldsymbol{\beta}_k + \boldsymbol{\varepsilon} \tag{3.1}$$

where $\mathbf{Y}$ is a $n \times 1$ vector of response y_i's, $\mathbf{X}_k$ is an $n \times p_k$ matrix corresponding to the kth group, $\boldsymbol{\beta}_k$ is a coefficient vector of size p_k for the kth group variable, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of errors ε_i for $i = 1, \ldots, n$ and $k = 1, \ldots, K$. Additionally, ε_i's are assumed to be statistically independent with mean 0 and standard deviation σ. Also, we assume that $||\beta_k||_2 \neq 0$ for $1 \leq k \leq k_0$ and $\boldsymbol{\beta}_k \equiv 0$ for $k \geq k_0$ where $|| \cdot ||_2$ is the ℓ_2 norm.

Statisticians have proposed regularized estimation techniques which can achieve variable selection and parameter estimation simultaneously. These methods stemmed from Lasso by Tibshirani [41] which combines a loss function and a penalty function. In general, we can summarize an objective function based on a loss and a penalty as follows.

$$L(\mathbf{Y}, \mathbf{X}, \boldsymbol{\beta}) + \lambda \sum_{k=1}^{K} p_{\lambda_k}(\boldsymbol{\beta}_k) \tag{3.2}$$

We review various techniques by combining L_2, L_1, or a rank-based loss L_{RB} as $L(\mathbf{Y}, \mathbf{X}, \boldsymbol{\beta})$, and $||\boldsymbol{\beta}_k||_2$, $||\boldsymbol{\beta}_k||_1$, $||\boldsymbol{\beta}_k||_\infty$, or SCAD as the penalty function $p_{\lambda_k}(\boldsymbol{\beta}_k)$ with or without adaptivity tuning parameter λ_k.

3.2 Loss Functions for Grouped Model

The loss functions for grouped model are the same as those for the non-grouped linear regression model. We detect the group feature of the given data with a grouped version of penalty functions.

We use the same loss functions as follows. Let $\mathbf{X}$ be the horizontally concatenated design matrix with $\mathbf{X_1}, \ldots, \mathbf{X_K}$ and $\boldsymbol{\beta}$ the corresponding vertically stacked coefficient parameters with $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$. That is $\mathbf{X} = (\mathbf{X_1}, \ldots, \mathbf{X_K})$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_K')'$.

1. The L_2 loss is written as the square of ℓ_2 norm of residuals

$$Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 \tag{3.3}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2. \tag{3.4}$$

2. The L_1 loss is the ℓ_1 norm of residuals

$$Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}|y_i - \mathbf{x}_i^T\boldsymbol{\beta}| \tag{3.5}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}|\varepsilon_i|. \tag{3.6}$$

3. The rank-based loss function L_{RB} [32, 20, 44] is

$$Loss_{RB}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) = \frac{1}{n}\sum_{i<j} b_{ij}|(y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (y_j - \mathbf{x}_j^T\boldsymbol{\beta})| \tag{3.7}$$

$$= \frac{1}{n}\sum_{i<j} b_{ij}|\varepsilon_i - \varepsilon_j|. \tag{3.8}$$

The choice of the loss function determines the robustness of estimation under the existence of outliers when we solve a linear regression problem (2.1).

3.3 Grouped Variable Selection via Regularization

3.3.1 Penalty Functions for Grouped Variables

Several group regularized penalty functions are proposed including group SCAD, group ℓ_1, group ℓ_2 and group ℓ_∞. We compare group ℓ_1 and group ℓ_2 penalty functions.

The penalty functions for grouped variable selection are defined as follows. For a d-dimensional vector $\mathbf{v} = (v_1, \ldots, v_d)$, we use the following norms of $\mathbf{v}$.

$$||\mathbf{v}||_2 = \Big(\sum_{k=1}^{d} v_k^2 \Big)^{1/2} \tag{3.9}$$

$$||\mathbf{v}||_1 = \sum_{k=1}^{d} |v_k| \tag{3.10}$$

1. The ℓ_2 group Lasso penalty for the kth group is

$$p_{\lambda_k,2}(\boldsymbol{\beta}) = \lambda_k ||\boldsymbol{\beta}_k||_2. \tag{3.11}$$

   The ℓ_2 group Lasso penalty has been used for group variable selection by Yuan and Lin [46]. It detects or deselects the entire group at once. We can achieve group adaptivity by choosing λ_k as $1/||\tilde{\boldsymbol{\beta}}_k||_2$ with initial estimates $\tilde{\boldsymbol{\beta}}_k$, $k = 1, \ldots, K$.

2. The ℓ_1 group Lasso penalty for the k-th group is

$$p_{\lambda_k,1}(\boldsymbol{\beta}) = \lambda_k ||\boldsymbol{\beta}_k||_1. \tag{3.12}$$

   Miakonkana et al. [32] proposed both group-wise and element-wise adaptive ℓ_1 Lasso penalty $\sum_{k=1}^{K} \sum_{j=1}^{p_k} \lambda_{kj} |\boldsymbol{\beta}_{kj}|$. The nature of this formula achieves both within-group sparsity and between-group sparsity.

The penalty function expresses group adaptivity with nontrivial group tuning parameter λ_k's. However, the penalty function excludes group adaptivity when all λ_k's are 1. Depending on λ_k for each group k, we can express group adaptivity or not.

The group adaptive ℓ_2 penalty function is the sum of all K penalties.

$$P_{\lambda,ag\ell_2}(\boldsymbol{\beta}) = \lambda \sum_{k=1}^{K} p_{\lambda_k,2}(\boldsymbol{\beta}) \tag{3.13}$$

$$= \lambda \sum_{k=1}^{K} \lambda_k ||\boldsymbol{\beta}_k||_2 \tag{3.14}$$

The group non-adaptive ℓ_2 penalty function is the sum of all K penalties with $\lambda_k = 1$ for all $k = 1, \ldots, K$.

$$P_{\lambda,g\ell_2}(\boldsymbol{\beta}) = \lambda \sum_{k=1}^{K} p_{1,2}(\boldsymbol{\beta}) \tag{3.15}$$

$$= \lambda \sum_{k=1}^{K} ||\boldsymbol{\beta}_k||_2 \tag{3.16}$$

The group adaptive ℓ_1 penalty function is the sum of all K penalties.

$$P_{\lambda,ag\ell_1}(\boldsymbol{\beta}) = \lambda \sum_{k=1}^{K} p_{\lambda_k}(\boldsymbol{\beta}) \tag{3.17}$$

$$= \lambda \sum_{k=1}^{K} \lambda_k ||\boldsymbol{\beta}_k||_1 \tag{3.18}$$

The group non-adaptive ℓ_1 penalty function is the sum of all K penalties with $\lambda_k = 1$ for all $k = 1, \ldots, K$.

$$P_{\lambda,g\ell_1}(\boldsymbol{\beta}) = \lambda \sum_{k=1}^{K} p_{1,1}(\boldsymbol{\beta}) \tag{3.19}$$

$$= \lambda \sum_{k=1}^{K} ||\boldsymbol{\beta}_k||_1 \tag{3.20}$$

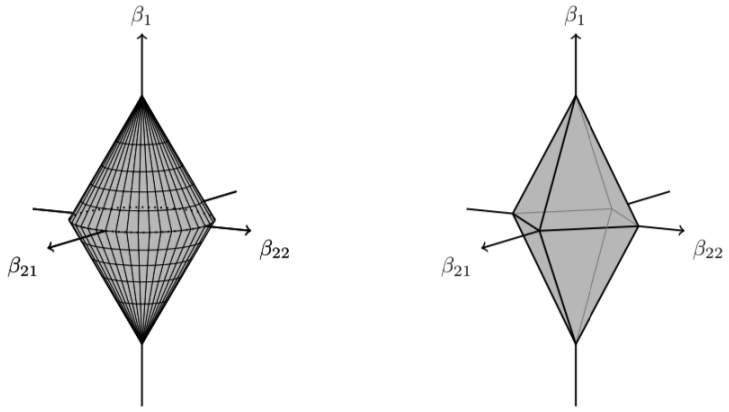

Figure 3.1: The unit ball for the group ℓ_2 norm (left) and the group ℓ_1 norm (right) : $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ where $\boldsymbol{\beta}_1 = \beta_1, \boldsymbol{\beta}_2 = (\beta_{21}, \beta_{22})$

3.3.2 Geometry of Penalty Functions

The differences between group penalty functions can be shown with their geometry. Figure 3.1 has two grouped vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ where $\boldsymbol{\beta}_1$ is in $\mathbb{R}$ and $\boldsymbol{\beta}_2$ is in $\mathbb{R}^2$. The unit ball for the group ℓ_1 is

$$\{(\beta_1, \beta_{21}, \beta_{22}) \in \mathbb{R}^3 : |\beta_1| + |\beta_{21}| + |\beta_{22}| \leq 1\} \tag{3.21}$$

and the unit ball for the group ℓ_2 is

$$\{(\beta_1, \beta_{21}, \beta_{22}) \in \mathbb{R}^3 : |\beta_1| + (\beta_{21}^2 + \beta_{22}^2)^{1/2} \leq 1\}. \tag{3.22}$$

We can have the minimum of the LS or RB loss functions at the edges or vertices of the balls. An ℓ_1 ball can meet at a vertex with a high chance to minimize the loss functions. However, an ℓ_2 ball meets with the circular edges to optimize the loss functions when $\boldsymbol{\beta}_1 = 0$ and $\boldsymbol{\beta}_2 \neq 0$. The group ℓ_2 penalty can obtain the estimated beta with all nonzero component values, but the group ℓ_1 penalty can estimate some zero values for a true parameter with all nonzero components. The group ℓ_2 penalty can achieve only the sparsity between groups. The group ℓ_1 penalty can accommodate the sparsity inside each group as well as between groups.

	group ℓ_2	group ℓ_1
L_2	$LS\_g\ell_2$	$LS\_g\ell_1$
L_1	$LAD\_g\ell_2$	$LAD\_g\ell_1$
L_{RB}	$RB\_g\ell_2$	$RB\_g\ell_1$

Table 3.1: Combinations of Loss and Penalty

	adaptive group ℓ_2	adaptive group ℓ_1
L_2	$LS\_ag\ell_2$	$LS\_ag\ell_1$
L_1	$LAD\_ag\ell_2$	$LAD\_ag\ell_1$
L_{RB}	$RB\_ag\ell_2$	$RB\_ag\ell_1$

Table 3.2: Combinations of Loss and Group Adaptive Penalty

3.4 Objective Functions for Grouped Variables

We can build 6 objective functions for group variable selection for multiple linear regression in Table 3.1 by combining three different loss functions in Section 3.2 and three different penalty functions in Section 3.3. Also, we consider their adaptive versions listed in Table 3.2.

We understand RB as a special case of LAD by generating $n(n-1)/2$ data pairs from the original data pair as follows.

$$x_{ij} = x_i - x_j \quad \text{and} \quad y_{ij} = y_i - y_j \tag{3.23}$$

for $i, j = 1, \ldots, n$ and $ij = 1, \ldots, n(n-1)/2$.

3.4.1 LS-based Objective Functions

The objective function $Q_{LS\_g\ell_2}(\boldsymbol{\beta})$ is the group Lasso objective function by Yuan and Lin [46]. The Lasso technique for multiple linear regression by Tibshirani [41] used the L_2 loss function which minimizes the residual sum of squares. It was extended to group linear regression model by modifying the penalty function with $||\boldsymbol{\beta}_k||_2$ to select group variables as in Yuan and Lin [46]. It pursued to select relevant groups rather than only individual variables and to deselect irrelevant groups by penalizing grouped parameters together and estimate parameters. The

objective function of Yuan and Lin [46] is equivalent to the following formula.

$$Q_{LS\_g\ell_2}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda,g\ell_2}(\boldsymbol{\beta}) \tag{3.24}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right)^2 + \frac{\lambda}{n}\sum_{k=1}^{K}||\boldsymbol{\beta}_k||_2 \tag{3.25}$$

Wang and Leng [42] modeled adaptive group Lasso by combining different tuning parameters for different groups. It is equivalent to minimize

$$Q_{LS\_ag\ell_2}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda,ag\ell_2}(\boldsymbol{\beta}) \tag{3.26}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right)^2 + \frac{\lambda}{n}\sum_{k=1}^{K}\lambda_k||\boldsymbol{\beta}_k||_2. \tag{3.27}$$

It was shown that the adaptive group Lasso estimates the true model consistently and that the estimator satisfies the oracle property.

Similarly, we can define objective functions by combining $P_{\lambda,g\ell_1}(\boldsymbol{\beta})$ and $P_{\lambda,ag\ell_1}(\boldsymbol{\beta})$. Apparently, non-adaptive penalty turns out to be the original Lasso in Tibshirani [41], that is, $Q_{LS\_g\ell_1}(\boldsymbol{\beta}) = Q_{LS\_\ell_1}(\boldsymbol{\beta})$.

$$Q_{LS\_g\ell_1}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda,g\ell_1}(\boldsymbol{\beta}) \tag{3.28}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right)^2 + \frac{\lambda}{n}\sum_{k=1}^{K}||\boldsymbol{\beta}_k||_1 \tag{3.29}$$

This does not detect a group feature. However, the group adaptive penalty function can obtain a group structure.

$$Q_{LS\_ag\ell_1}(\boldsymbol{\beta}) = Loss_{LS}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda,ag\ell_1}(\boldsymbol{\beta}) \tag{3.30}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right)^2 + \frac{\lambda}{n}\sum_{k=1}^{K}\lambda_k||\boldsymbol{\beta}_k||_1 \tag{3.31}$$

But still this is equivalent to a special case of adaptive non-grouped Lasso method in Section 2.5.1. We have a group adaptive penalty λ_k, $k = 1, \ldots, K$ instead of element-wise adaptive penalty λ_j, $j = 1, \ldots, p$.

3.4.2 LAD-Based Objective Functions

The L_1 loss function is a special case of quantile regression when $\tau = 0.5$. L_1 loss minimization was suggested to overcome the shortcoming of the least square minimization method. Bang and Jhun [4] used the quantile regression loss function for group variable selection method with the group ℓ_∞ penalty for multiple linear regression model. Lilly and Billor [29] proposed the least absolute deviation (LAD) group variable selection method with L_1 loss with group ℓ_1 penalty for multiple linear regression model.

$$Q_{LAD\_g\ell_1}(\boldsymbol{\beta}) = Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda, g\ell_1}(\boldsymbol{\beta}) \tag{3.32}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left|y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right| + \frac{\lambda}{n}\sum_{k=1}^{K}||\boldsymbol{\beta}_k||_1 \tag{3.33}$$

This is equivalent to non-grouped LAD Lasso model, however, the group feature can be captured by the LAD loss function with group adaptive penalty function.

$$Q_{LAD\_ag\ell_1}(\boldsymbol{\beta}) = Loss_{LAD}(\boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) + P_{\lambda, ag\ell_1}(\boldsymbol{\beta}) \tag{3.34}$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left|y_i - \sum_{k=1}^{K}\mathbf{x}_{ik}^T\boldsymbol{\beta}_k\right| + \frac{\lambda}{n}\sum_{k=1}^{K}\lambda_k||\boldsymbol{\beta}_k||_1 \tag{3.35}$$

3.4.3 L_{RB} Loss Functions

Johnson and Feng [20] Wang and Li [44] proposed a rank-based variable selection method for multiple linear regression model without a group structure.

$$\frac{1}{n}\sum_{i=1}^{n} b_{ij}|e_i - e_j| + n\sum_{j=1}^{p} p_\lambda(|\beta_j|) \tag{3.36}$$

where p_λ is the SCAD penalty function for variable selection. Wang and Li added the weighted factor b_{ij} for the robustness in predictors whereas the model of Johnson and Feng [20] has $b_{ij} = 1$ for all $1 \leq i, j \leq n$.

We combine the RB loss function with $P_{\lambda,ag\ell_2}(\boldsymbol{\beta})$, $P_{\lambda,g\ell_2}(\boldsymbol{\beta})$, $P_{\lambda,ag\ell_1}(\boldsymbol{\beta})$, and $P_{\lambda,g\ell_1}(\boldsymbol{\beta})$ in Section 3.3.1. Similarly to the non-grouped case, we generate new data set x_{ij} and y_{ij} for $ij = 1, \ldots, n(n-1)/2$. Then, we perform LAD group Lasso with or without adaptivity for $n(n-1)/2$ generated data pairs to obtain RB group Lasso estimates.

3.5 Choice of Tuning Parameter λ

We use 10-fold cross-validation (CV) and Schwarz information criterion (SIC) to choose the optimal tuning parameter λ for a grouped penalized model. We choose λ that minimizes the prediction errors for the test sets. Bang and Jhun [4] used a robust SIC for LAD. Kim and Koivunen [23] used BIC for RB which is equivalent to SIC. We calculate SIC for each loss function as follows.

$$\text{SIC}_{LS} = \ln \frac{1}{n} \sum_i ||e_i||^2 + \frac{1}{n} df \cdot \ln n \tag{3.37}$$

$$\text{SIC}_{LAD} = \ln \frac{1}{n} \sum_i |e_i| + \frac{1}{2n} df \cdot \ln n \tag{3.38}$$

$$\text{SIC}_{RB} = \ln \frac{1}{N} \sum_{i<j} b_{ij} |e_i - e_j| + \frac{1}{2n} df \cdot \ln n \tag{3.39}$$

where $N = n(n-1)/2$, n is the number of observations, and df is the model size. We choose the best tuning parameter λ which minimizes SIC for each loss function.

Chapter 4

Functional Linear Model

The mathematical foundation for functional data analysis assumes that we observe curve data. Functional data analysis is concerned with functional observations defined on a set $\mathcal{T}$. Even though the data has the form of repeated measurements at finitely many points only, the nature of some data has functional structure in it. For example, we observe temperature over the year at discrete time points, but, the temperature data can be understood as a continuous functional data. We use functional data analysis tools for converting raw discrete data to functional continuous data. That is, we smooth out the sparse and disconnected observed data with the use of kernel, B-spline, Fourier, polynomial, or wavelet basis as described in Ramsay and Silverman [38].

With pre-processed data, we develop the analysis technique over a Hilbert space, $L^2(\mathcal{T})$, which is a complete infinite-dimensional normed space of functions on $\mathcal{T}$.

We model the observed data functions as single entities rather than a sequence of individual observations. We convert a sequence of individual discrete observations $x_s = x(t_s)$ for $s = 0, 1, \ldots, m$ to a function x with values $x(t)$ for an arbitrary $t \in \mathcal{T}$ by smoothing. Linear algebra plays a roll to represent functions as a linear combination of basis functions over a Hilbert space of $L^2(\mathcal{T})$ integrable functions on $\mathcal{T}$ with $L^2(\mathcal{T})$ norm given by

$$||x||_2^2 = \langle x, x \rangle = \int_{\mathcal{T}} x^2(t)dt \tag{4.1}$$

where $\langle \cdot, \cdot \rangle$ is the inner product on this functional space defined by

$$\langle x, y \rangle = \int_{\mathcal{T}} x(t)y(t)dt \quad \text{for any } x \text{ and } y \in L^2(\mathcal{T}). \tag{4.2}$$

For any basis ϕ_k of $L^2(\mathcal{T})$, we have

$$x(t) = \sum_{\ell=1}^{\infty} \langle x, \phi_\ell \rangle \phi_\ell(t) \tag{4.3}$$

by the Karhunen-Loéve expansion where the convergence is in $L^2(\mathcal{T})$ with probability one. For a fixed number d of basis, we may approximate $x(t)$ with

$$x(t_s) \approx \sum_{\ell=1}^{d} c_\ell \phi_\ell(t_s). \tag{4.4}$$

The coefficients are obtained by minimizing the following sum of squares

$$\min \sum_{s=1}^{m} \left(x(t_s) - \sum_{\ell=1}^{d} c_\ell \phi_\ell(t_s) \right)^2. \tag{4.5}$$

Let $\boldsymbol{\phi}$ be the $m \times d$ matrix $\{\phi_\ell(t_s)\}_{\ell,s}$, $\mathbf{x}$ the discrete data vector $\{x(t_s)\}_s = \{x_s\}_s$, and $\mathbf{c}$ the coefficient vector $\{c_\ell\}_\ell$ of length d. Then the previous approximation equation (4.4) becomes

$$\mathbf{x} \approx \mathbf{c}^T \boldsymbol{\phi}. \tag{4.6}$$

If we write Equation (4.4) on a continuous functional space,

$$x(t) \approx \mathbf{c}^T \boldsymbol{\phi}(t) \tag{4.7}$$

where the choice of the basis ϕ_ℓ and the dimension d is crucial, which might depend on the discrete data. Usually we use Fourier basis for periodic data, B-spline basis for non-periodic data, and wavelet bases for the data without differentiability requirements. The optimal dimension d for approximation is found by minimizing the mean squared error as discussed by Horváth

and Kokoszka [16]. Ramsay and Silverman [38] discussed about the methods for choosing the number of basis to approximate. A larger d improves the approximation in Equation (4.5), but increases the variance of the estimated statistics. However, this dissertation will use a fixed number of basis without optimization for the dimension d. The choice of basis for functional principal component technique is discussed in Horváth and Rice [17] using a hypothesis test method. The L^2 norm of the second derivative of $x(t)$ is related to the curvature of the function curve $x(t)$. A smaller L^2 norm $||x''(t)||_2^2$ obtains a smoother functional curve. We can control smoothness of the functional version $x(t)$ of the discrete data $x_s = x(t_s)$ by adding the term $||x''(t)||_2^2$ to the least square minimization Equation (4.5). Considering all these facts, we can determine a proper basis and the number of basis with a proper smoothness.

Several functional linear regression model can be considered as follows. We can combine a scalar or a functional response with scalar or functional predictors to set up a functional linear model. First, a model with both functional response and functional predictors with scalar parameters can be described by

$$y_i(t) = \alpha + \sum_{j=1}^{p} \beta_j X_{ij}(t) + \varepsilon. \tag{4.8}$$

In this dissertation, we focus on the model with functional predictors and a scalar response. Consider a functional multiple linear regression model with p functional predictors and a continuous scalar response defined by Equation (1.1).

$$y_i = \alpha + \sum_{j=1}^{p} \int_{\mathcal{T}} X_{ij}(t)\beta_j(t)dt + \varepsilon_i, \quad i = 1, \ldots, n,$$

where y_i is a scalar and $X_{ij}(t)$'s on $\mathcal{T}$, the support of functional covariates, are L^2 integrable and independent with each other, $\beta_j(t)$'s are functional parameters which are also L^2 integrable, and $\varepsilon_i \overset{iid}{\sim} F$, where F is some distribution with finite Fisher information. The functional linear regression model in Equation (1.1) has been studied to estimate the parameter function $\beta(t)$'s using L_2 [38], L_1 [29], and L_{RB} [8] loss functions.

4.1 Penalized Functional Linear Model for Variable Selection

We reviewed the regularized methods for group variable selection in ordinary multiple regression model in Chapter 3. The ideal technique for group variable selection is expected to select significant groups and estimate parameters simultaneously which performs well under a diverse range of distribution of errors in response and high leverage predictor data. These can be achieved by combining a loss function to take care of non-normal errors in response space, a penalty term to select sparse meaningful group variable, and weight in loss function to control high leverage in predictor space. Moreover, if we add group adaptivity, the parameter estimators acquire the oracle property. In addition, we consider the smoothness of estimated parameter functions by controlling the concavity with their second derivatives. We formulate Equation (1.1) using a basis expansion on a functional space in Section 4.1.1. Section 4.1.2 expresses the functional linear model in Equation (1.1) based on a transformed basis to include the second derivative of the parameter functions. Section 4.1.3 sets up for the penalized functional linear model with L_2 loss and group adaptive ℓ_2 penalty in Gertheiss et al. [12].

4.1.1 Functional Linear Model as a Multiple Linear Regression

We express a multiple functional linear model as an original multiple linear regression model with a group structure using basis expansion. We express the functional model in Equation (1.1) as a discretized form over $\{t_1, \ldots, t_m\} \in \mathcal{T}$ with an appropriate kind of basis set $\{\phi(t)\}$ and an appropriate finite number of basis d. With the finite basis $\phi_{j1}, \ldots, \phi_{jd}$, the parameter function $\beta_j(t)$ can be written as a finite dimensional approximation

$$\beta_j(t) \approx \sum_{\ell=1}^{d} c_{j\ell}\phi_\ell(t). \tag{4.9}$$

Then we can approximate the integration in (1.1) as

$$\int_{\mathcal{T}} X_{ij}(t)\beta_j(t)dt \approx \sum_{s=1}^{m} X_{ij}(t_s)\beta_j(t_s)(t_s - t_{s-1}) \tag{4.10}$$

$$\approx \sum_{\ell}\left(\sum_{s} X_{ij}(t_s)\phi_\ell(t_s)\delta_s\right)c_{j\ell} \tag{4.11}$$

$$= \sum_{\ell} \Phi_{ij\ell}c_{j\ell} \tag{4.12}$$

$$= \mathbf{\Phi}_{ij}^T\mathbf{c}_j \tag{4.13}$$

where $i = 1, \ldots, n$, $j = 1, \ldots, p$, $\delta_s = t_s - t_{s-1}$, $\mathbf{c}_j = (c_{j1}, \ldots, c_{jd})^T$, $\mathbf{\Phi}_{ij} = (\Phi_{ij1}, \ldots, \Phi_{ijd})^T$ and $\Phi_{ij\ell} = \sum_s X_{ij}(t_s)\phi_\ell(t_s)\delta_s$.

The discretized version of our model is written as

$$y_i = \alpha + \sum_{j=1}^{p} \mathbf{\Phi}_{ij}^T\mathbf{c}_j + \varepsilon_i, \quad i = 1, \ldots, n \tag{4.14}$$

which is a grouped multiple linear regression model with p groups, d predictors in each group, and n observations. The functional linear regression model in Equation (1.1) becomes a discrete grouped regression model to estimate grouped parameters $\mathbf{c}_j$'s for $j = 1, \ldots, p$.

4.1.2 Penalty with the Second Derivatives of the Coefficient Functions

We express the penalty terms by changing basis to include the second derivative of parameter functions. Then we model the penalized functional linear model for variable selection with loss functions and penalty functions. We discuss about the L_{RB} loss with l_2 penalty for the functional regression model in section 5. Gertheiss et al. [12] proposed a functional variable selection method which estimates coefficient functions and controls functional smoothness simultaneously about (1.1) by using the sparsity-smoothness penalty $J(f_j)$in Meier et al. [31].

$$J(f_j) = \lambda_1\sqrt{||f_j||_n^2 + \lambda_2 I^2(f_j)} \tag{4.15}$$

where $I^2(f_j) = \int (f_j''(x))^2 dx$ measures the smoothness of f_j.

The penalty term is a functional version of ℓ_2 group Lasso penalty including the second derivative of parameter functions $\beta_j(t)$, $j = 1, \ldots, p$ to control smoothness. The objective function is

$$\sum_{i=1}^{n} \left(y_i - \alpha - \sum_{j=1}^{p} \mathbf{\Phi}_{ij}^T \mathbf{c}_j \right)^2 + \sum_{j=1}^{p} P_{\lambda,\varphi}(\beta_j) \tag{4.16}$$

where

$$P_{\lambda,\varphi}(\beta_j) = \lambda(||\beta_j||_2^2 + \varphi||\beta_j''||_2^2)^{1/2}, \tag{4.17}$$

$||\cdot||^2$ is the functional L^2 norm in Equation (4.1), and $\beta_j''(t) = d^2\beta_j(t)/dt^2$.

Here we focus on the functional 2-norm penalty term in Equation (4.17) which is analogous to the group ℓ_2 penalty function in Equation (3.11). λ is the tuning parameter which controls sparseness. If λ is zero, we do not drop any variables and include all predictors as significant in the estimated model. However, a huge λ value can shrink down the estimated model by selecting none of predictors as significant. The smoothness of the parameter functions is controlled by φ. As the smoothness parameter φ increases, the estimated parameter functions $\widehat{\beta}_j(t)$ achieve more smoother curve. For examples, $\varphi = 0$ case will make the most fluctuating estimated curve. If φ is large enough, we may expect to have straight lines as estimated functional parameters. The suitable λ and φ are chosen via K-fold cross-validation by minimizing the prediction error of the estimation. We follow the setting in Gertheiss et al. [12] to restructure the penalty function $P_{\lambda,\varphi}(\beta_j)$ in Equation (4.17). For each $j = 1, \ldots, p$, we define the inner products Ψ_j between the basis functions and the inner product Ω_j between the second derivative of basis functions for a adequately chosen d degree of freedom as follows.

$$\Psi_j = \begin{pmatrix} \langle \phi_{j1}, \phi_{j1} \rangle & \langle \phi_{j1}, \phi_{j2} \rangle & \cdots & \langle \phi_{j1}, \phi_{jd} \rangle \\ \langle \phi_{j2}, \phi_{j1} \rangle & \langle \phi_{j2}, \phi_{j2} \rangle & \cdots & \langle \phi_{j2}, \phi_{jd} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_{jd}, \phi_{j1} \rangle & \langle \phi_{jd}, \phi_{j2} \rangle & \cdots & \langle \phi_{jd}, \phi_{jd} \rangle \end{pmatrix} \tag{4.18}$$

$$\Omega_j = \begin{pmatrix} \langle \phi''_{j1}, \phi''_{j1} \rangle & \langle \phi''_{j1}, \phi''_{j2} \rangle & \cdots & \langle \phi''_{j1}, \phi''_{jd} \rangle \\ \langle \phi''_{j2}, \phi''_{j1} \rangle & \langle \phi''_{j2}, \phi''_{j2} \rangle & \cdots & \langle \phi''_{j2}, \phi''_{jd} \rangle \\ \cdots & \cdots & \ddots & \cdots \\ \langle \phi''_{jd}, \phi''_{j1} \rangle & \langle \phi''_{jd}, \phi''_{j2} \rangle & \cdots & \langle \phi''_{jd}, \phi''_{jd} \rangle \end{pmatrix} \tag{4.19}$$

$$C_{\varphi,j} = \Psi_j + \varphi \Omega_j \tag{4.20}$$

The linear combination of Ψ_j and Ω_j can span L^2 functional space and reduce the functional penalty term (4.17) as

$$P_{\lambda,\varphi}(\beta_j) = \lambda(||\beta_j||_2^2 + \varphi||\beta''_j||_2^2)^{1/2} \tag{4.21}$$

$$= \lambda(\mathbf{c}_j^T C_{\varphi,j} \mathbf{c}_j)^{1/2}. \tag{4.22}$$

Furthermore, $C_{\varphi,j}$ can be written as

$$C_{\varphi,j} = L_{\varphi,j} L_{\varphi,j}^T \tag{4.23}$$

where $L_{\varphi,j}$ is a non-singular lower triangular matrix by Cholesky decomposition. From Equations (4.16) and (4.23), we have

$$\mathbf{\Phi}_{ij}^T \mathbf{c}_j = \mathbf{\Phi}_{ij}^T I_{d\times d} \mathbf{c}_j \tag{4.24}$$

$$= \mathbf{\Phi}_{ij}^T (L_{\varphi,j}^T)^{-1} L_{\varphi,j}^T \mathbf{c}_j \tag{4.25}$$

$$= (L_{\varphi,j}^{-1} \mathbf{\Phi}_{ij})^T L_{\varphi,j}^T \mathbf{c}_j \tag{4.26}$$

$$= \widetilde{\mathbf{\Phi}}_{ij}^T \widetilde{\mathbf{c}}_{\varphi,j} \tag{4.27}$$

where $\widetilde{\mathbf{\Phi}}_{ij} = L_{\varphi,j}^{-1}\mathbf{\Phi}_{ij}$ and $\widetilde{\mathbf{c}}_{\varphi,j} = L_{\varphi,j}^{T}\mathbf{c}_j$. The penalty function (4.22) with the second derivatives is

$$P_{\lambda,\varphi}(\beta_j) = \lambda(||\beta_j||_2^2 + \varphi||\beta_j''||_2^2)^{1/2} \tag{4.28}$$

$$= \lambda(\mathbf{c}_j^T C_{\varphi,j}\mathbf{c}_j)^{1/2} \tag{4.29}$$

$$= \lambda||\widetilde{\mathbf{c}}_{\varphi,j}||_2. \tag{4.30}$$

4.1.3 Gertheiss' Objective Function for L_2 Loss and ℓ_2 Penalty for Various Adaptivity

The functional linear regression model (1.1) can be written as

$$y_i = \alpha + \sum_{j=1}^{p} \widetilde{\mathbf{\Phi}}_{ij}^T\widetilde{\mathbf{c}}_{\varphi,j} + \varepsilon_i \quad \text{for } i = 1,\ldots,n. \tag{4.31}$$

Thus, Gertheiss et al. [12] uses the following objective function to estimate $\hat{\alpha}$ and $\hat{\widetilde{\mathbf{c}}}_j$ using the group Lasso method.

$$Q_{L_2,\ell_2}(\alpha,\widetilde{\mathbf{c}}_j) = \sum_{i=1}^{n}\left(y_i - \alpha - \sum_{j=1}^{p}\widetilde{\mathbf{\Phi}}_{ij}^T\widetilde{\mathbf{c}}_{\varphi,j}\right)^2 + \sum_{j=1}^{p}\lambda||\widetilde{\mathbf{c}}_{\varphi,j}||_2 \tag{4.32}$$

The above equations (4.31) and (4.32) do not reflect the group adaptivity. Gertheiss et al. define the adaptive penalization similar to the adaptive group Lasso by Zou [47] by adding weights to control the contribution of the jth parameter function $\beta_j(t)$ and its second derivative $\beta_j''(t)$ for each $j = 1,\ldots,p$. Their adaptive penalty function is

$$P_{a,\lambda,\varphi}(\beta_j) = \lambda(w_j||\beta_j||_2^2 + \varphi v_j||\beta_j''||_2^2)^{1/2} \tag{4.33}$$

where the weights w_j's and v_j's are chosen depending on data. The tuning parameter λ controls the entire penalty function and φ controls the concavity of estimated parameter functions $\beta_j(t)$'s. However, the weights w_j and v_j reflect the size of the jth parameter function $\beta_j(t)$ for each j by defining $w_j = 1/||\hat{\beta}_j||_2$ and $v_j = 1/||\hat{\beta}_j''||_2$ for a set of estimated coefficient function $\hat{\beta}_j(t)$. These weights help to detect the correct nonzero coefficient functions by giving

smaller weights to meaningful covariates and larger weights to insignificant covariates. These two weights are not tuning parameters but fixed values. Thus the adaptiveness does not change add any difficulty to computation to optimize an objective function. Specifically, the penalty function (4.33) can be rewritten in the following way. Similarly, we define $C_{a,\varphi,j}$ as

$$C_{a,\varphi,j} = \sqrt{w_j}\Psi_j + \varphi\sqrt{v_j}\Omega_j \tag{4.34}$$

and rewrite using Cholesky decomposition as

$$C_{a,\varphi,j} = L^a_{\varphi,j}(L^a{}_{\varphi,j})^T \tag{4.35}$$

with a non-signular lower triangular matrix $L_{a,\varphi,j}$. Equation (4.16) can be written as

$$\begin{aligned}
\mathbf{\Phi}_{ij}^T\mathbf{c}_j &= \mathbf{\Phi}_{ij}^T I_{d\times d}\mathbf{c}_j && (4.36)\\
&= \mathbf{\Phi}_{ij}^T(L_{a,\varphi,j}^T)^{-1}L_{a,\varphi,j}^T\mathbf{c}_j && (4.37)\\
&= (L_{a,\varphi,j}^{-1}\mathbf{\Phi}_{ij})^T L_{a,\varphi,j}^T\mathbf{c}_j && (4.38)\\
&= \widetilde{\mathbf{\Phi}}_{a,ij}^T\widetilde{\mathbf{c}}_{a,\varphi,j} && (4.39)
\end{aligned}$$

where $\widetilde{\mathbf{\Phi}}_{a,ij} = L_{a,\varphi,j}^{-1}\mathbf{\Phi}_{ij}$ and $\widetilde{\mathbf{c}}_{a,\varphi,j} = L_{a,\varphi,j}^T\mathbf{c}_j$.

The most general expression for the functional linear model (1.1) becomes

$$y_i = \alpha + \sum_{j=1}^{p}\widetilde{\mathbf{\Phi}}_{a,ij}^T\widetilde{\mathbf{c}}_{a,\varphi,j} + \varepsilon_i \quad \text{for } i = 1,\ldots,n. \tag{4.40}$$

The group adaptivity ℓ_2 penalty function will give the following objective function with L^2 loss.

$$Q_{a,L_2,\ell_2}(\alpha,\widetilde{\mathbf{c}}_{\varphi,j}) = \sum_{i=1}^{n}\left(y_i - \alpha - \sum_{j=1}^{p}\widetilde{\mathbf{\Phi}}_{a,ij}^T\widetilde{\mathbf{c}}_{a,\varphi,j}\right)^2 + \sum_{j=1}^{p}\lambda||\widetilde{\mathbf{c}}_{a,\varphi,j}||_2 \tag{4.41}$$

We may apply different kinds of adaptivity by setting $w_j = 1$ for all $j = 1,\ldots,p$ or $v_j = 1$ for all $j = 1,\ldots,p$. If the initial coefficient functions have expressive difference on their sizes, but all coefficient functions have similar smoothness, we employ only w_j by setting $v_j = 1$

for $j = 1, \ldots, p$. On the other hand, we utilize only v_j by letting $w_j = 1$ for all $j = 1, \ldots, p$ in the case that we have similar sizes for all coefficient functions with larger variance on their smoothness.

In general, we use Equation (4.40) for the group adaptivity in functional linear regression model.

$$y_i = \alpha + \sum_{j=1}^{p} \widetilde{\mathbf{\Phi}}_{a,ij}^T \widetilde{\mathbf{c}}_{a,\varphi,j} + \varepsilon_i, \quad \text{for } i = 1, \ldots, n$$

We estimate the optimal α and $\widetilde{\mathbf{c}}_{\varphi,j}$ for $j = 1, \ldots, p$ in Equation (4.40) using the rank-based loss function, L_{RB}, in (2.20) with the group ℓ_2 in (3.11) penalty function in Chapter 5.

Chapter 5

Rank Based Group Variable Selection for Functional Linear Model

5.1 Introduction

We propose the rank-based group variable selection method for functional linear model. We combine the rank-based loss function L_{RB}, in Equatioin (2.20) and the group ℓ_2 penalty function in Equation (3.11) and finally define an L_{RB} objective function as follows using the group adaptive penalty function (4.33).

$$Q_{a,RB} = \frac{1}{n}\sum_{i<j}^{n} b_{ij}|\varepsilon_i - \varepsilon_j| + \sum_{j=1}^{p} P_{a,\lambda,\varphi}(\beta_j) \tag{5.1}$$

where

$$\begin{aligned} P_{a,\lambda,\varphi}(\beta_j) &= \lambda(w_j||\beta_j||_2^2 + \varphi v_j||\beta_j''||_2^2)^{1/2} \\ &= \lambda||\widetilde{\mathbf{c}}_{a,\varphi,j}||_2 \end{aligned}$$

with Equation (4.40)

$$y_i = \alpha + \sum_{j=1}^{p} \widetilde{\mathbf{\Phi}}_{a,ij}^T \widetilde{\mathbf{c}}_{a,\varphi,j} + \varepsilon_i, \quad \text{for } i = 1, \ldots, n$$

We use the same setting as Gertheiss et al.[12] except for the loss function. Then we apply the grouped multiple linear regression method with the RB loss functions with the group ℓ_2 penalty with or without adaptivity. The rank-based loss function in Equation (2.20) has the weights depending on predictors and can reduce the effect of data observations with high leverage.

Moreover, the nature of the rank-based loss can achieve the robustness under the existence of outliers in the response space.

We explain the weights b_{ij}'s and how to obtain λ and φ. We create the simulation data without outliers in both the predictor space and response space. Afterwards, we generate the contaminated data with the outliers in the response space and the observations with high leverage value in the predictor space.

The rank-based estimator estimates and selects $\beta(t)$ simultaneously. We obtain the optimal smoothness by choosing the best φ. With adaptivity, we can see the oracle property of the estimator of $\beta(t)$ in Chapter A.

5.2 Weights b_{ij}

The weights b_{ij} are from the pairwise difference data $x_{ij} = x_i - x_j$ and $y_{ij} = y_i - y_j$. We define as Equation (2.13)

$$b_{ij} = b(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)h(\mathbf{x}_j) \tag{5.2}$$

which degrades high leverage points, where

$$h(\mathbf{x}_i) = \min\left[1, \frac{b}{(\mathbf{x}_i - \hat{\mu})^T S^{-1}(\mathbf{x}_i - \hat{\mu})}\right] \tag{5.3}$$

with $(\hat{\mu}, S)$ being the robust minimum volume ellipsoid estimators of the location and spreadness, and b the 95th percetile of $\chi^2(p)$ for the number of predictors p, as in Wang and Li [44] and Miakonkana et al. [32]. We calculate the robust estimators, $(\hat{\mu}, S)$, using the MCD (Minimum Covariance Determinant) by ocvMcd() function in the R package robustbase.

5.3 Choosing Tuning Parameters λ and φ

We use cross-validation (CV) and Schwarz information criterion (SIC) to choose the optimal tuning parameter λ for a functional penalized model as we discussed about the grouped linear model in Section 3.5.

The K-fold cross-validation selects the optimal λ and φ for the L_2 loss function in Equation (4.41) and the rank-based loss function in Equation (5.1) with the group ℓ_2 penalty. We

split the data observations into K subsets randomly. For each subset, we estimate the parameters using the rest of $K-1$ subsets and then predict the response for one chosen subset based on the estimates by $K-1$ subsets. We choose the best tuning parameters such that they minimize the mean of the prediction error for the response over K subsets.

To optimize the tuning parameters using SIC, we use the following definitions.

$$\text{SIC}_{LS} = \ln \frac{1}{n} \sum_i ||e_i||^2 + \frac{1}{n} df \cdot \ln n$$
$$\text{SIC}_{LAD} = \ln \frac{1}{n} \sum_i |e_i| + \frac{1}{2n} df \cdot \ln n$$
$$\text{SIC}_{RB} = \ln \frac{1}{N} \sum_{i<j} b_{ij} |e_i - e_j| + \frac{1}{2n} df \cdot \ln n$$

where $N = n(n-1)/2$, n is the number of observations, and df is the model size.

For a simple simulation study with $p = 4$ predictor functions, we try to find the optimal λ by minimizing the root mean squared error of β, ($RMSE(\beta)$), since we know the true $\beta(t)$'s with CV.

5.4 Simulation Study

We generate sine-like functional predictors similarly as in Gerthesis [12] and generate the responses by adding errors from different kinds of distributions to the inner product between the coefficient functions $\beta(t)$'s and the functional predictors. We create the contaminated predictors which resembles the data with high leverage observations. We compare the results between LS, LAD, and RB loss with the group ℓ_2 penalties with and without optimization of smoothness.

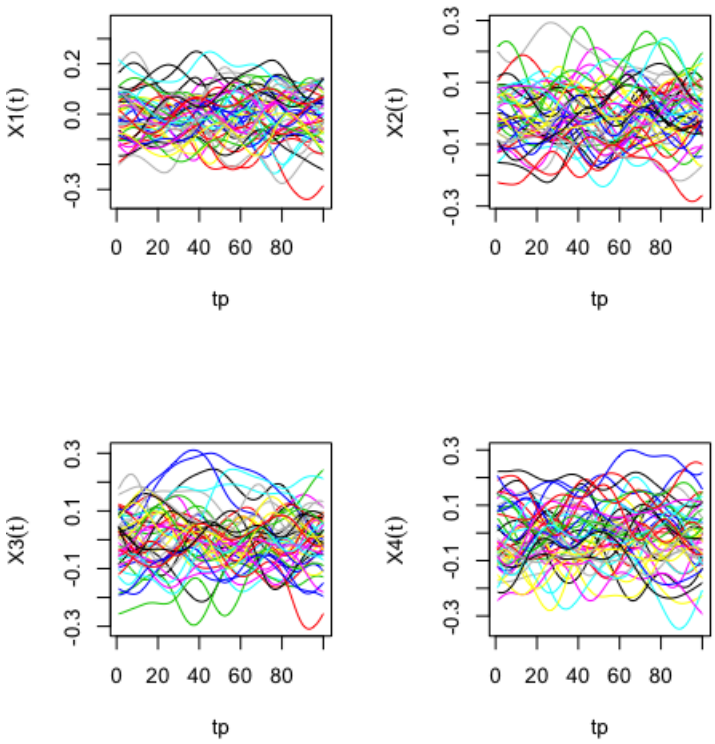

Figure 5.1: c_0: Predictor functions without contamination

5.4.1 Data Generation

Consider an example in which four functional covariates are observed at a set of 100 equidistant points in $(0, 100)$ for each sampling unit. Define for $i = 1, \ldots, n$, and $k = 1, \ldots, 4$,

$$x_{ik}(t) = \sum_{r=1}^{5} a_{ikr} \sin\Big(\frac{2\pi(5 - a_{ikr})}{150}\Big)t - m_{ikr}, k = 1, \ldots, 4 \tag{5.4}$$

$$y_i = \sum_{k=1}^{4} \int_0^{100} x_{i,k}(t)\beta_k(t)dt + \varepsilon_i, \quad i = 1, \ldots, n \tag{5.5}$$

where $a_{ikr} \sim U(0, 5)$, $m_{ikr} \sim U(0, 2\pi), i = 1, \ldots, n$, $k = 1, \ldots, 4$, $r = 1, \ldots, 5$ and $t \in [0, 100]$. Figure 5.1 shows the predictor functions. The true parameter functions $\beta_1(t)$ and $\beta_2(t)$ are γ distribution density curves with different stretches and $\beta_3(t) = \beta_4(t) = 0$ as shown in Figure 5.2.

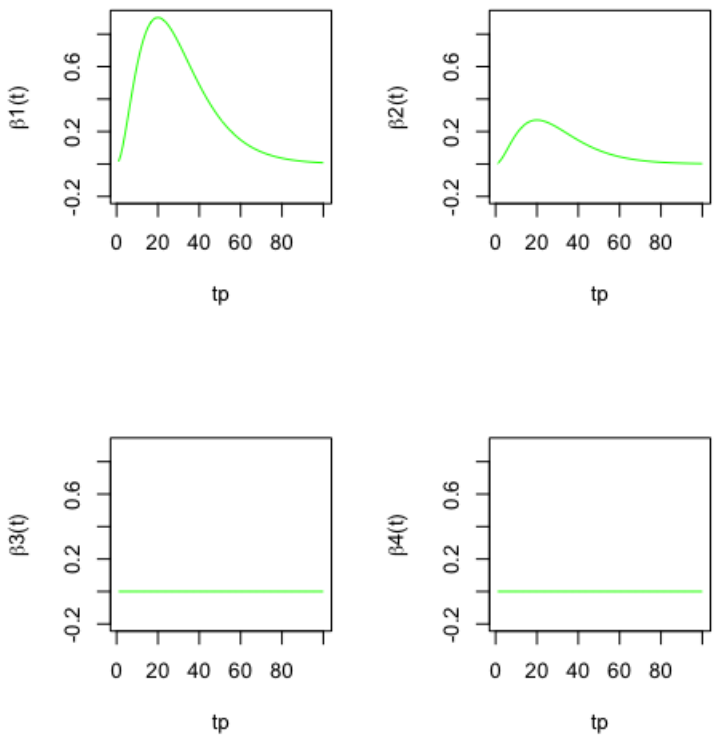

Figure 5.2: True parameter $\beta(t)$ curves

For the response y_i, we use $\varepsilon_i \sim N(0,1)$ (en) , t_3 (et3), and a mixed normal, $0.95N(0,1)+0.05N(0,10^2)$, (em) to compare the performance. We run the simulation 100 times. We discretize the functional simulation data with 10 B-spline basis as in Chapter 4, then compare the result based on the LS, LAD, and RB loss functions with group ℓ_2 penalty function. We will simulate the date with the sample size $n = 100$ and 150 for $p = 4$. We compare the result for $p = 10$ with $n = 100$.

To see the effect of the weights b_{ij}, we generate the contaminated data in the predictor space. We use the contamination criteria in Fraiman and Muniz [11]. We use three types of 15% contamination for each predictor function with the contamination size constant $M = 5$. The types are asymmetric, symmetric, and partial contaminations. They are generated by the following definitions.

- Asymmetric contamination(c_1):

$$z_{i,k}^{a}(t) = x_{i,k}(t) + cM$$

where $c \sim Bernoulli(0.15)$ and $M = 5$

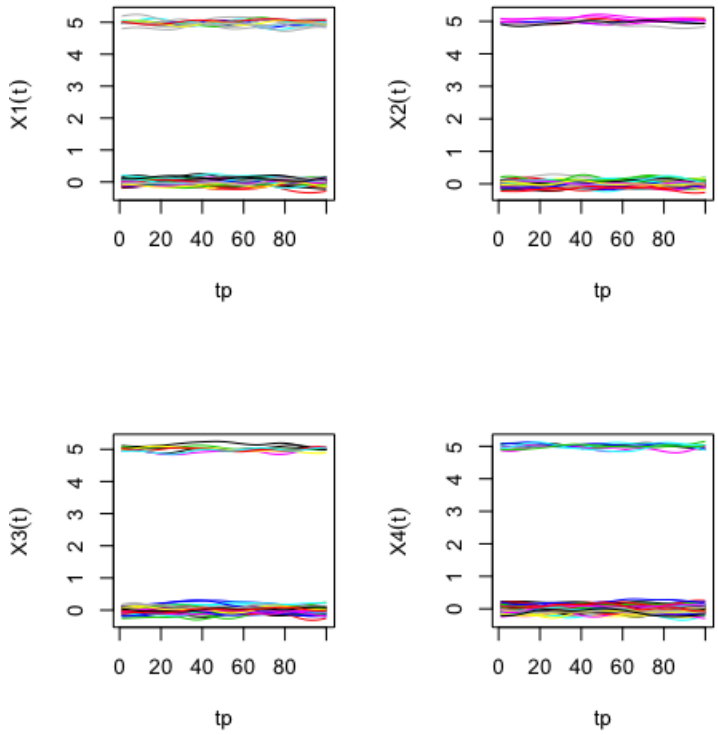

Figure 5.3: c_1: Predictor functions with 15% asymmetric contamination

- Symmetric contamination(c_2):

$$z_{i,k}^{s}(t) = x_{i,k}(t) + c\sigma M$$

where $c \sim Bernoulli(0.15)$, $M = 5$, and σ is a random variable independent of c which is 1 or -1 with probability 0.5.

- Partial contamination(c_3):

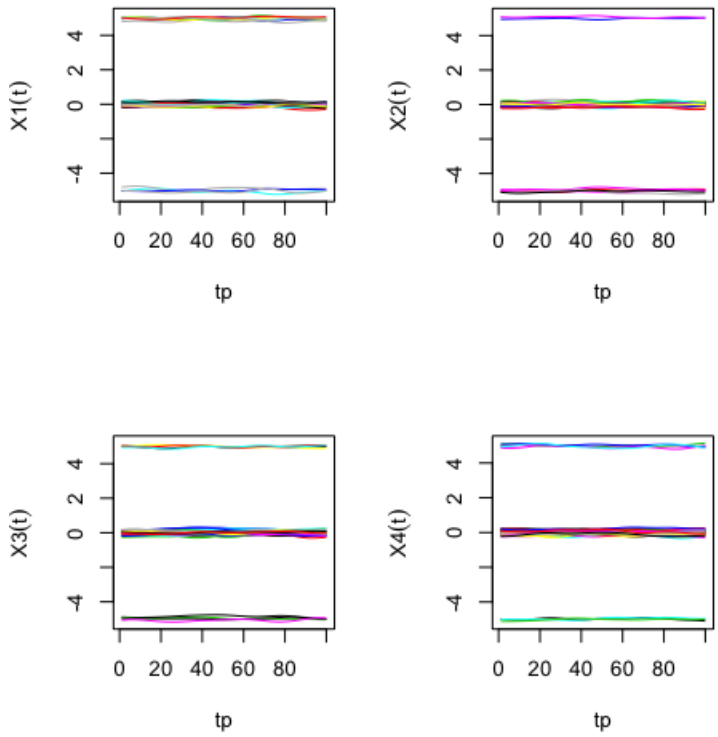

Figure 5.4: c_2: Predictor functions with 15% symmetric contamination

$$z_{i,k}^{p}(t) = \begin{cases} x_{i,k}(t) + c\sigma M & , t > T \\ x_{i,k}(t) & , t < T \end{cases}$$

where $T \sim U(0, 10)$, $c \sim Bernoulli(0.15)$, $M = 5$, and σ is a random variable independent of c which is 1 or -1 with probability 0.5.

- No contamination (c_0):

$$z_{i,k}^{no}(t) = x_{i,k}(t)$$

Thus, we consider asymmetric contamination in Figure 5.3, symmetric contamination in Figure 5.4, and partial contamination in Figure 5.5 contaminations in the x direction and three kinds of y direction errors, the standard normal (en) , t_3 (et3) and the mixed normal errors (em)

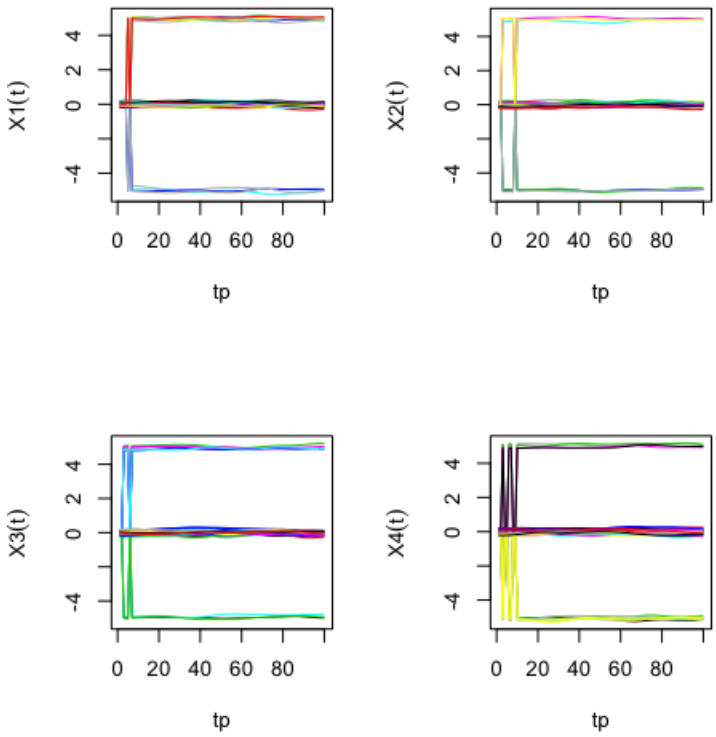

Figure 5.5: c_3: Predictor functions with 15% partial contamination

with box-plots as shown in Figure 5.6 . We compare the results between LS, LAD, and RB loss functions with different combinations of contaminations.

The following sections summarize the simulation results by comparing LS, LAD and RB methods under the different combinations of contaminations and with or without the optimization for the smoothness parameter φ. We search the RMSE (Root Mean Squared Error) of $\beta(t)$ to compare the performance between loss functions. The $RMSE(\beta)$ is defined as

$$RMSE(\beta) = \Big[\int (\hat{\beta}(t) - \beta(t))^2 dt \Big]^{1/2}$$

where $\hat{\beta}(t)$ is the estimated parameter function for the true parameter function $\beta(t)$.

5.4.2 Results on c_0 : No outliers in the predictor space

We start with the result on the data without x-direction contaminations and with the standard normal error (en), Huber mixed normal errors (em) and t_3 errors (et3) for $n = 50$ observations.

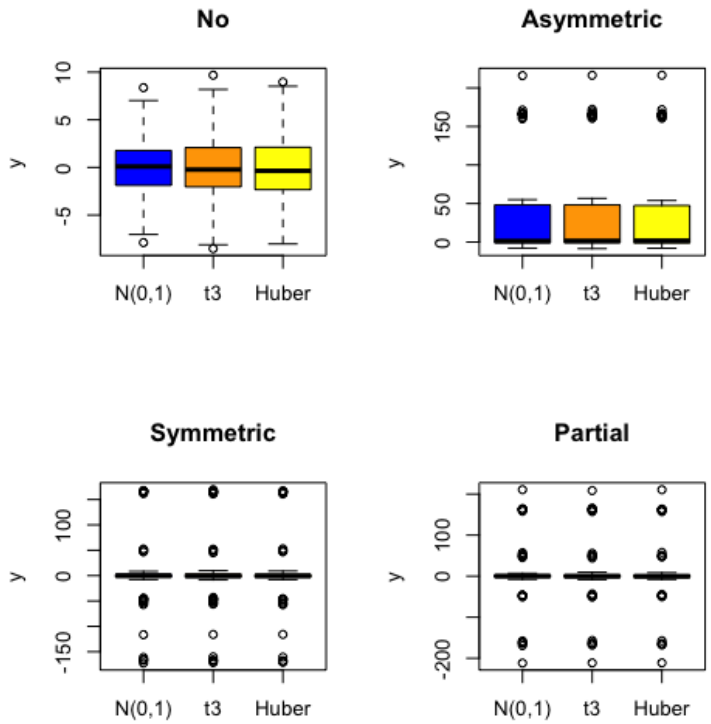

Figure 5.6: Boxplots of responses for all combinations of x and y-contaminations

Since we do not have the x contaminations, we use the weight $b_{ij} = 1$ for all $i, j = 1, \ldots, 50$. And we do not penalize the smoothness by letting $\varphi = 0$ for all cases. We optimize the tuning parameter λ by 10-fold cross-validation and by minimizing the root mean squared error of β ($RMSE(\beta)$). We run 100 simulations for each y error. We only compare LS and RB.

		LS						RB					
		x1	x2	x3	x4	Model.Size	Model.Error	x1	x2	x3	x4	Model.Size	Model.Error
c_0	en	1	1	0.34	0.41	2.75	0.0645	1	1	0.33	0.37	2.7	0.067
	em	1	0.93	0.48	0.33	2.74	0.109	1	1	0.37	0.39	2.76	0.072
	t_3	1	1	0.44	0.40	2.84	0.093	1	1	0.37	0.46	2.83	0.079
	Oracle	1	1	0	0	2	0	1	1	0	0	2	0

Table 5.1: Comparison under y outliers based on $RMSE(\beta)$

Table 5.1 shows the comparison between LS and RB methods under the presence of the outliers in the response space. LS performs better than RB method under the standard normal error since LS has the smaller model error. With the presence of outliers in the response space, RB estimates better than LS. RB and LS have 0.072 and 0.109 as the model errors, respectively. LS fails to detect the second variable in 7%, however, RB detects the second one as significant

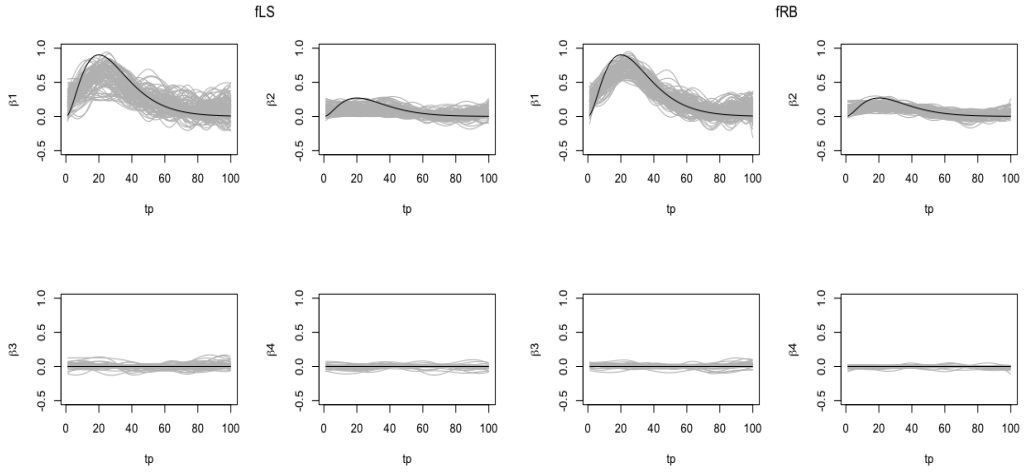

Figure 5.7: Estimated $\beta(t)$ under Huber mixed normal errors by $RMSE(\beta)$

		LS						RB					
		x1	x2	x3	x4	Model.Size	Model.Error	x1	x2	x3	x4	Model.Size	Model.Error
c_0	en	1	1	0.63	0.68	3.31	0.0658	1	1	0.54	0.62	2.674	0.0648
	em	1	0.95	0.62	0.52	3.19	0.1102	1	1	0.59	0.51	3.1	0.0765
	t_3	1	0.98	0.61	0.59	3.18	0.0959	1	1	0.53	0.58	3.11	0.0818

Table 5.2: Comparison under y outliers based on CV

variable in 100%. The rank-based method also has a smaller model error for t_3 error with a smaller model size. This simulation result says the rank-based method performs better under the presence of response outliers. Figure 5.7 shows the performance difference between LS and RB under Huber mixed normal errors by choosing λ which minimizes $RMSE(\beta)$. RB method gives better precision with narrower estimated clouds than LS. The estimation using the cross-validation in Table 5.2 gives similar results with smaller model errors by the rank-based method.

cont		LS							RB						
		x1	x2	x3	x4	Model.Size	Ave(RMSE(β))	Sd(RMSE(β))	x1	x2	x3	x4	Model.Size	Ave(RMSE(β))	Sd(RMSE(β))
c_0	en	1	1	0.680	0.730	3.410	1.501	0.439	1	1	0.780	0.780	3.560	1.546	0.374
	em	1	0.990	0.640	0.700	3.330	2.255	0.491	1	1	0.790	0.800	3.590	1.833	0.462
	et3	1	1	0.700	0.650	3.350	1.995	0.416	1	1	0.760	0.800	3.560	1.910	0.520

Table 5.3: y-contaminated data with optimization for λ and φ

Table 5.3 has the result 100 simulations for $n = 100$ observations without x contaminations with considering the smoothness parameter φ and the weight for the rank-based loss, b_{ij}. We optimize both λ and φ using 10-fold cross-validation. The case with standard normal error has a smaller average of $RMSE(\beta)$ for the estimated β under LS method. However, RB method obtains smaller average of $RMSE(\beta)$ than LS method.

5.4.3 Results under x and y outliers when $p = 4$

This section shows the results under outliers in both the predictor space and the response space. Firstly, we optimize the smoothness penalty φ for non-contaminated data set using LS loss function. This case should give an ideal φ estimation and it can be applied to all other loss functions, LAD and RB and the data sets with outliers. After choosing an optimal φ, we use the estimated nonzero φ for all models and choose tuning parameter λ using 10-fold CV (cross-validation) and SIC (Schwarz information criterion). Since we know the true β, we use the weights for $||\beta(t)||_2$ and $||\beta''(t)||_2$ as the true $\beta(t)$'s for adaptive estimations. Adapt0 estimates $\beta(t)$'s without any adaptation. Adapt1 uses the adaptivity for $||\beta(t)||_2$ and Adapt2 uses the adaptivity for both $||\beta(t)||_2$ and $||\beta''(t)||_2$. We compare results based on different loss functions, x and y outliers, and different adaptivities using two tuning parameter choosing methods, 10-fold CV and SIC with different numbers of observations. The data sets are same to the previous simulation results under $RMSE(\beta)$ and CV. The data sets have the same 4 predictor functions and the same y outliers with the same true $\beta(t)$ in 5.2. However, we use 4 degree of freedom for basis instead of 10 and compare the results for 100 and 150 observations to check the oracle property. The following sections include the results for x outliers, c_0, c_1, c_2, and c_3 of 100 simulations. We compare the performances between different loss functions based on $RMSE(\beta)$ and the average model sizes. We also check the oracle property based on $RMSE(\beta)$ and the average model sizes by comparing the results between different sample sizes. The table below provide the average model size (df), the mean $RMSE(\beta)$ (m(β)), and the standard deviation of $RMSE(\beta)$ (sd(β)).

A. x contamination : c_0

Tables 5.4 through 5.9 show the results of 100 simulations for $n = 100$ and $n = 150$ observations without x contaminations for three different loss functions under y outliers. LS performs best under the standard normal error in y direction as expected since LS works best under a normal error in both CV and SIC with or without adaptivity. Under normal errors, RB after LS performs better than LAD. However, RB performs best under mixed

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.42	2.007	0.134	3.34	2.11	0.141	3.46	2.049	0.133
100	em	3.19	2.423	0.358	3.5	2.207	0.18	3.5	2.127	0.189
100	et3	3.31	2.209	0.253	3.45	2.172	0.174	3.46	2.166	0.238
150	en	3.43	2.005	0.155	3.51	2.075	0.148	3.54	2.045	0.157
150	em	3.27	2.169	0.257	3.39	2.056	0.139	3.58	2.05	0.187
150	et3	3.45	2.121	0.201	3.44	2.111	0.191	3.54	2.112	0.2

Table 5.4: c_0 Adapt0 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.27	2.029	0.149	2.53	2.15	0.183	2.38	2.09	0.216
100	em	1.8	2.66	0.904	2.25	2.168	0.159	2.13	2.134	0.209
100	et3	2.18	2.262	0.347	2.39	2.181	0.191	2.23	2.178	0.292
150	en	2.14	1.983	0.149	2.31	2.082	0.181	2.13	2.012	0.157
150	em	1.94	2.242	0.373	2.17	2.044	0.119	2.07	1.993	0.091
150	et3	2.06	2.116	0.199	2.25	2.083	0.17	2.08	2.04	0.131

Table 5.5: c_0 Adapt0 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.049	0.147	2	2.19	0.104	2	2.115	0.153
100	em	1.97	2.439	0.362	2.02	2.24	0.146	2.01	2.189	0.19
100	et3	2	2.259	0.259	2	2.26	0.168	2	2.219	0.217
150	en	2.01	2.017	0.153	2	2.166	0.139	2	2.046	0.167
150	em	1.99	2.224	0.269	2.01	2.149	0.122	2.01	2.045	0.16
150	et3	1.99	2.181	0.244	2	2.187	0.154	2	2.127	0.197

Table 5.6: c_0 Adapt1 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.065	0.18	2	2.239	0.201	2	2.085	0.195
100	em	1.65	2.924	0.771	2.01	2.319	0.218	2	2.204	0.255
100	et3	1.97	2.504	0.453	2	2.349	0.275	2	2.305	0.317
150	en	2	2.009	0.157	2	2.155	0.183	2	2.022	0.162
150	em	1.82	2.5	0.532	2	2.18	0.188	2	2.041	0.172
150	et3	1.98	2.276	0.36	2	2.236	0.225	2	2.108	0.23

Table 5.7: c_0 Adapt1 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.05	2.173	0.136	2	2.211	0.105	2	2.217	0.118
100	em	1.98	2.427	0.312	2	2.229	0.105	2.05	2.259	0.158
100	et3	1.99	2.297	0.212	2	2.253	0.125	2.02	2.272	0.173
150	en	2.11	2.113	0.165	2	2.204	0.091	2.07	2.178	0.137
150	em	1.99	2.268	0.177	2	2.205	0.09	2.06	2.194	0.122
150	et3	2	2.243	0.168	2	2.221	0.113	2.06	2.23	0.143

Table 5.8: c_0 Adapt2 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.096	0.175	2	2.265	0.141	2	2.089	0.191
100	em	1.64	2.715	0.503	2.01	2.305	0.166	2	2.198	0.235
100	et3	1.96	2.364	0.36	2	2.313	0.167	2	2.243	0.285
150	en	2	2.042	0.163	2.01	2.262	0.117	2	2.027	0.164
150	em	1.83	2.366	0.43	2	2.26	0.13	2	2.027	0.161
150	et3	1.99	2.221	0.28	2	2.293	0.147	2	2.102	0.225

Table 5.9: c_0 Adapt2 by SIC

normal errors and t_3 errors and then LAD and LS in order. CV always estimated larger model sizes than SIC. That is, CV chooses a smaller tuning parameter λ than SIC.

With adaptivities, the estimated model sizes approach to the true number of nonzero predictor functions. Moreover, we can check the oracle property with adaptivities, Adapt1 and Adapt2, using all different loss functions under all y outliers by both CV and SIC. All results report the smaller $RMSE(\beta)$ with the larger number of observations, $n = 150$. The objective functions have numerous local minima to choose a tuning parameter λ especially by CV. SIC makes sudden decreases in its value while λ increases. Thus, we can compare the performance between Adapt1 and Adapt2 by SIC. Most of cases using any loss functions, Adapt2 gives better results with smaller $RMSE(\beta)$. In all adaptivities, three loss functions have the order, RB, LAD, and LS in terms of $RMSE(\beta)$ under the mixed normal error (em).

B. x contamination : c_1

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.98	2.079	0.095	2.74	3.23	0.01	3.93	2.248	0.259
100	em	3.75	2.423	0.393	2.81	3.23	0.011	3.95	2.442	0.403
100	et3	3.95	2.217	0.233	2.87	3.232	0.009	3.92	2.45	0.374
150	en	3.97	1.996	0.083	2.76	3.211	0.134	3.87	2.108	0.214
150	em	3.94	2.151	0.257	2.74	3.207	0.154	3.85	2.103	0.251
150	et3	4	2.082	0.153	2.65	3.218	0.097	3.79	2.175	0.294

Table 5.10: c_1 Adapt0 by CV

Tables 5.10 through 5.15 show the results of 100 simulations for $n = 100$ and $n = 150$ observations with asymmetric contaminations (c_1) in predictors for three different loss functions under y outliers. In all adaptivities, LAD performs worst under all y error using both

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.79	2.125	0.149	2.08	3.229	0.019	3.49	2.492	0.439
100	em	2.55	2.843	0.419	2.01	3.234	0.011	3.29	2.676	0.605
100	et3	2.95	2.583	0.427	2.06	3.233	0.012	3.45	2.836	0.745
150	en	3.76	2.028	0.086	3.05	2.521	0.495	3.41	2.238	0.33
150	em	2.73	2.662	0.412	3.05	2.446	0.468	3.18	2.219	0.422
150	et3	3.38	2.322	0.302	2.94	2.558	0.49	3.19	2.388	0.501

Table 5.11: c_1 Adapt0 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.29	2.095	0.059	2	3.21	0.011	2.63	2.234	0.231
100	em	2.1	2.359	0.288	2	3.208	0.018	2.6	2.386	0.33
100	et3	2.24	2.193	0.145	2	3.211	0.012	2.62	2.413	0.308
150	en	2.24	2.035	0.04	2	3.094	0.304	2.55	2.111	0.194
150	em	2.12	2.131	0.147	2	3.156	0.217	2.36	2.088	0.189
150	et3	2.11	2.091	0.091	2	3.175	0.18	2.39	2.181	0.216

Table 5.12: c_1 Adapt1 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.093	0.058	2	3.215	0.008	2.07	2.257	0.257
100	em	2	2.377	0.294	2	3.216	0.009	2.03	2.368	0.304
100	et3	2	2.191	0.148	2	3.217	0.01	2.04	2.458	0.478
150	en	2	2.035	0.043	2.14	2.585	0.388	2.05	2.113	0.207
150	em	2	2.155	0.169	2.03	2.628	0.408	2	2.097	0.206
150	et3	2	2.092	0.091	2.08	2.642	0.404	2.01	2.23	0.279

Table 5.13: c_1 Adapt1 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.42	2.108	0.04	2	3.288	0.009	2.59	2.271	0.168
100	em	2.13	2.294	0.268	2	3.288	0.01	2.62	2.361	0.229
100	et3	2.31	2.161	0.095	2	3.291	0.01	2.63	2.355	0.255
150	en	2.4	2.094	0.029	2	3.264	0.166	2.71	2.185	0.144
150	em	2.23	2.154	0.098	2	3.268	0.153	2.46	2.175	0.16
150	et3	2.26	2.127	0.071	2	3.278	0.112	2.52	2.24	0.16

Table 5.14: c_1 Adapt2 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.104	0.033	2	3.294	0.007	2.12	2.294	0.2
100	em	2	2.278	0.221	2	3.295	0.008	2.07	2.383	0.257
100	et3	2	2.158	0.09	2	3.296	0.008	2.04	2.432	0.452
150	en	2	2.092	0.025	2.23	2.66	0.418	2.28	2.222	0.221
150	em	2	2.165	0.125	2.02	2.718	0.394	2.05	2.202	0.158
150	et3	2	2.119	0.055	2.05	2.783	0.409	2.13	2.295	0.297

Table 5.15: c_1 Adapt2 by SIC

CV and SIC with or without adaptivity. In most cases, LS performs better than other loss functions. The simulations are based on the same weights of β and β'' for the true parameter functions $\beta(t)$. We could try to use the estimated parameter for each loss function to calculate the weights for adaptivities. The estimation under Adapt2 does not improve the result comparing with Adapt1. This might come from a wrongly chosen φ and the weight for the smoothness since φ is estimated under the non-contaminated case in x and y. However, in each adaptivity, we can check the oracle property for each loss function. Also, in Adapt0 and Adapt1, RB performs best with the mixed normal error (em). Adapt1 gives better results than Adapt0 if we compare the results limited on each loss function.

C. x contamination : c_2

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.98	2.059	0.087	2.63	3.232	0.013	3.92	2.23	0.225
100	em	3.86	2.395	0.382	2.7	3.236	0.01	3.95	2.383	0.368
100	et3	3.97	2.193	0.226	2.78	3.234	0.012	3.96	2.412	0.375
150	en	4	2.004	0.09	2.92	3.23	0.006	3.71	2.139	0.268
150	em	3.95	2.131	0.221	2.87	3.195	0.18	3.66	2.109	0.227
150	et3	3.98	2.082	0.154	2.86	3.231	0.005	3.69	2.209	0.337

Table 5.16: c_2 Adapt0 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.69	2.138	0.187	2.04	3.229	0.014	3.47	2.452	0.423
100	em	2.44	2.863	0.396	2.01	3.234	0.011	3.22	2.651	0.65
100	et3	3.03	2.573	0.423	2.04	3.23	0.027	3.54	2.937	0.783
150	en	3.81	2.029	0.079	3.21	2.453	0.443	2.38	2.079	0.244
150	em	2.92	2.578	0.421	3.03	2.48	0.49	2.05	2.016	0.163
150	et3	3.41	2.271	0.317	2.89	2.554	0.508	2.23	2.119	0.327

Table 5.17: c_2 Adapt0 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.26	2.09	0.059	2	3.212	0.008	2.61	2.217	0.206
100	em	2.04	2.354	0.298	2	3.212	0.011	2.47	2.356	0.297
100	et3	2.24	2.182	0.141	2	3.214	0.01	2.48	2.362	0.266
150	en	2.2	2.038	0.053	2	3.061	0.327	2.6	2.103	0.195
150	em	2.12	2.129	0.148	2	3.102	0.307	2.49	2.089	0.19
150	et3	2.15	2.094	0.109	2	3.128	0.241	2.4	2.2	0.252

Table 5.18: c_2 Adapt1 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.087	0.058	2	3.216	0.008	2.04	2.254	0.259
100	em	2.01	2.379	0.3	2	3.217	0.009	2.02	2.394	0.348
100	et3	2	2.181	0.143	2	3.218	0.009	2.02	2.438	0.431
150	en	2	2.035	0.043	2.11	2.557	0.369	2.04	2.098	0.207
150	em	2	2.148	0.167	2.05	2.558	0.384	2.01	2.078	0.193
150	et3	2	2.094	0.094	2.06	2.671	0.399	2	2.21	0.259

Table 5.19: c_2 Adapt1 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.28	2.105	0.034	2.02	3.291	0.009	2.59	2.255	0.17
100	em	2.11	2.284	0.261	2	3.291	0.009	2.44	2.342	0.227
100	et3	2.34	2.16	0.092	2	3.292	0.01	2.55	2.335	0.222
150	en	2.27	2.095	0.032	2	3.185	0.282	2.2	2.23	0.143
150	em	2.21	2.153	0.1	2.01	3.197	0.294	2.18	2.203	0.104
150	et3	2.19	2.123	0.063	2	3.251	0.175	2.11	2.245	0.15

Table 5.20: c_2 Adapt2 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.103	0.032	2	3.296	0.007	2.17	2.317	0.231
100	em	2.01	2.275	0.221	2	3.296	0.008	2.04	2.382	0.242
100	et3	2	2.156	0.088	2	3.297	0.008	2.05	2.422	0.412
150	en	2	2.092	0.025	2.17	2.679	0.425	2.03	2.264	0.166
150	em	2	2.164	0.127	2.06	2.747	0.388	2	2.262	0.136
150	et3	2	2.121	0.059	2.1	2.8	0.419	2	2.316	0.178

Table 5.21: c_2 Adapt2 by SIC

Tables 5.16 through 5.21 show the results of 100 simulations for $n = 100$ and $n = 150$ observations with symmetric contaminations (c_2) in predictors for three different loss functions under y outliers. The results of c_2 are similar to them under c_1. LAD performs worst in terms of $RMSE(\beta)$ even if it estimates the model size properly in most cases. The results do not have improvement with Adapt2 from Adapt1. However, in each loss function, we can see the oracle property holds. For Adapt0 and Adapt1, RB performs better than LS under the mixed normal error (em).

D. x contamination : c_3

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.99	2.001	0.083	3.05	3.402	0.011	3.81	2.178	0.248
100	em	3.89	2.396	0.437	3.11	3.404	0.011	3.81	2.266	0.337
100	et3	3.96	2.15	0.218	3.1	3.404	0.01	3.78	2.284	0.328
150	en	3.97	1.949	0.091	2.38	3.369	0.011	3.96	2.191	0.291
150	em	3.91	2.089	0.2	2.44	3.357	0.108	3.92	2.145	0.251
150	et3	3.98	2.036	0.152	2.41	3.359	0.066	3.94	2.268	0.3

Table 5.22: c_3 Adapt0 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	3.72	2.057	0.139	2.09	3.401	0.014	3.52	2.502	0.521
100	em	2.72	2.864	0.489	2.06	3.403	0.014	3.21	2.437	0.563
100	et3	3.26	2.508	0.46	2.1	3.403	0.013	3.5	2.751	0.627
150	en	3.78	2.001	0.091	3.23	2.542	0.556	3.52	2.225	0.335
150	em	2.85	2.618	0.458	3.17	2.536	0.571	3.28	2.202	0.395
150	et3	3.63	2.267	0.328	3.2	2.539	0.569	3.31	2.376	0.485

Table 5.23: c_3 Adapt0 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.12	2.051	0.06	2	3.382	0.013	2.63	2.17	0.212
100	em	2.05	2.334	0.297	2	3.383	0.014	2.62	2.252	0.256
100	et3	2.12	2.161	0.158	2	3.388	0.015	2.64	2.355	0.347
150	en	2.26	2.014	0.046	2	3.273	0.299	2.93	2.07	0.235
150	em	2.05	2.124	0.161	2	3.267	0.301	2.84	2.069	0.22
150	et3	2.15	2.088	0.144	2	3.298	0.252	2.78	2.172	0.227

Table 5.24: c_3 Adapt1 by CV

Tables 5.22 through 5.27 show the results of 100 simulations for $n = 100$ and $n = 150$ observations with partial contaminations (c_3) in predictors for three different loss functions under y outliers. The results of c_3 are similar to them under c_1 and c_2. LAD performs

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.05	0.058	2	3.375	0.009	2.13	2.224	0.352
100	em	2	2.371	0.321	2	3.376	0.011	2.01	2.242	0.287
100	et3	2	2.156	0.152	2	3.381	0.011	2.02	2.277	0.318
150	en	2	2.013	0.043	2.07	2.699	0.449	2.04	2.05	0.208
150	em	2	2.159	0.186	2.02	2.712	0.438	2.01	2.026	0.179
150	et3	2	2.084	0.104	2.06	2.78	0.493	2.04	2.158	0.257

Table 5.25: c_3 Adapt1 by SIC

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2.2	2.123	0.039	2	3.463	0.01	2.6	2.17	0.145
100	em	2.12	2.296	0.233	2	3.465	0.011	2.61	2.212	0.141
100	et3	2.23	2.186	0.109	2	3.469	0.012	2.54	2.257	0.219
150	en	2.32	2.106	0.032	2	3.35	0.272	3.21	2.081	0.208
150	em	2.13	2.171	0.115	2	3.404	0.199	3.06	2.077	0.2
150	et3	2.18	2.143	0.132	2	3.396	0.207	2.9	2.166	0.218

Table 5.26: c_3 Adapt2 by CV

n	y	ls.df	ls.m(β)	ls.sd(β)	lad.df	lad.m(β)	lad.sd(β)	rb.df	rb.m(β)	rb.sd(β)
100	en	2	2.121	0.036	2	3.457	0.008	2.22	2.259	0.28
100	em	2	2.301	0.224	2	3.458	0.009	2.03	2.238	0.169
100	et3	2	2.18	0.106	2	3.462	0.01	2.09	2.285	0.261
150	en	2	2.103	0.029	2.02	2.893	0.453	2.31	2.162	0.217
150	em	2	2.184	0.132	2.05	2.881	0.441	2.12	2.197	0.193
150	et3	2	2.134	0.062	2.04	2.968	0.451	2.15	2.253	0.23

Table 5.27: c_3 Adapt2 by SIC

worst in terms of $RMSE(\beta)$ even if it estimates the model size properly in most cases. The results do not have improvement with Adapt2 from Adapt1. However, in each loss function, we can see the oracle property holds. We have better results in model size estimation and $RMSE(\beta)$ in all three methods. For Adapt0 and Adapt1, RB performs better than LS under the mixed normal error (em).

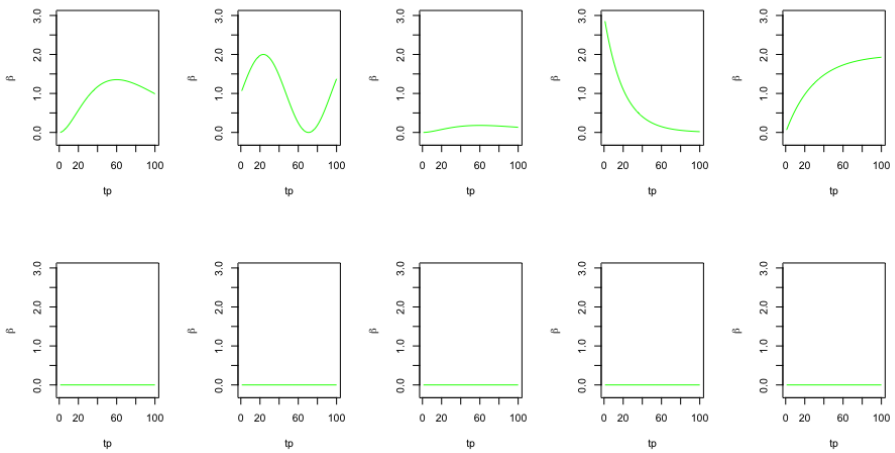

Figure 5.8: True $\beta(t)$'s when $p = 10$

5.4.4 Results under x and y outliers when $p = 10$

We consider 10 predictor functions with 5 nonzero and 5 zero coefficient functions with 4 cubic spline basis functions. The true coefficient functions are in Figure 5.8

We use the same kind of, sine-like, predictor functions with the same three kinds of contaminations in predictor space. We consider the standard normal error (en), the mixed normal error (em) and t_3 error in the response space. For all cases, we use an estimated constant φ using non-contaminated data. For the fixed estimated φ, we find the tuning parameter λ using CV with LS and RB loss functions when the number of observation is $n = 100$.

Table 5.28, 5.29 and 5.30 have the results for $p = 10$ with $n = 100$ observation under 100 simulations using CV. Without adaptivity, (Adapt0), RB outperforms LS in all cases in terms of average model sizes and $RMSE(\beta)$ except c_0 with the standard normal error (en). In Adapt1, RB performs better in c_0 cases without the standard normal errors (en). For other x outliers, RB has smaller $RMSE(\beta)$ with larger model sizes. In Adapt2, RB outperforms LS under all

cont		LS												RB										
x	y	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)
c0	en	1	1	1	1	1	0.87	0.85	0.84	0.87	0.80	9.23	3.39	0.26	1	1	1	1	1	0.95	0.87	0.99	0.91	0.88	9.60	3.49	0.26
c0	em	1	1	1	1	1	0.86	0.58	0.82	0.79	0.82	8.87	4.76	0.92	1	1	1	1	1	0.90	0.81	0.93	0.87	0.85	9.36	3.78	0.33
c0	et3	1	1	1	1	1	0.82	0.75	0.89	0.85	0.82	9.13	3.95	0.50	1	1	1	1	1	0.87	0.83	0.91	0.90	0.88	9.39	3.88	0.42
c1	en	1	1	1	1	1	1.00	1.00	1.00	0.98	1.00	9.98	5.98	0.20	1	1	1	1	1	0.97	1.00	1.00	1.00	1.00	9.97	3.79	0.37
c1	em	1	1	1	1	1	0.98	0.99	0.99	1.00	1.00	9.96	6.51	0.54	1	1	1	1	1	0.93	0.99	1.00	1.00	1.00	9.92	4.13	0.53
c1	et3	1	1	1	1	1	1.00	0.99	0.99	0.99	1.00	9.97	6.17	0.38	1	1	1	1	1	0.98	1.00	1.00	0.99	1.00	9.97	4.28	0.59
c2	en	1	1	1	1	1	1.00	1.00	1.00	1.00	0.99	9.99	5.84	0.22	1	1	1	1	1	0.95	1.00	1.00	1.00	1.00	9.95	3.62	0.35
c2	em	1	1	1	1	1	1.00	1.00	1.00	0.99	1.00	9.99	6.29	0.57	1	1	1	1	1	0.99	1.00	1.00	1.00	1.00	9.99	3.82	0.43
c2	et3	1	1	1	1	1	1.00	1.00	1.00	0.99	1.00	9.99	6.00	0.38	1	1	1	1	1	0.98	1.00	1.00	1.00	1.00	9.98	4.03	0.53
c3	en	1	1	1	1	1	1.00	1.00	1.00	1.00	0.99	9.99	5.49	0.21	1	1	1	1	1	0.68	0.98	0.98	0.63	1.00	9.27	3.98	0.49
c3	em	1	1	1	1	1	0.99	0.99	1.00	0.99	0.99	9.96	6.00	0.56	1	1	1	1	1	0.75	0.99	0.99	0.60	0.96	9.29	4.39	0.55
c3	et3	1	1	1	1	1	1.00	0.99	0.99	1.00	1.00	9.98	5.66	0.41	1	1	1	1	1	0.71	0.99	0.97	0.61	0.99	9.27	4.48	0.73

Table 5.28: $p = 10$,$n = 100$ with Adapt0 by CV

cont		LS												RB										
x	y	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)
c0	en	1	1	1	1	1	0.00	0.00	0.00	0.00	0.00	5.00	3.22	0.22	1	1	1	1	1	0.00	0.01	0.00	0.00	0.00	5.01	3.25	0.24
c0	em	1	1	1	1	1	0.00	0.00	0.00	0.00	0.00	5.00	4.41	0.95	1	1	1	1	1	0.00	0.00	0.00	0.00	0.00	5.00	3.39	0.27
c0	et3	1	1	1	1	1	0.00	0.00	0.00	0.00	0.00	5.00	3.68	0.45	1	1	1	1	1	0.01	0.01	0.00	0.00	0.01	5.03	3.50	0.33
c1	en	1	1	1	1	1	0.00	0.11	0.04	0.02	0.07	5.24	6.52	0.15	1	1	1	1	1	0.00	0.18	0.31	0.20	0.27	5.96	3.49	0.36
c1	em	1	1	1	1	1	0.03	0.06	0.03	0.02	0.10	5.24	6.78	0.38	1	1	1	1	1	0.00	0.26	0.39	0.25	0.38	6.28	3.76	0.51
c1	et3	1	1	1	1	1	0.01	0.12	0.05	0.01	0.05	5.24	6.58	0.30	1	1	1	1	1	0.22	0.35	0.15	0.15	0.27	6.14	3.94	0.98
c2	en	1	1	1	1	1	0.04	0.06	0.13	0.15	0.04	5.42	6.37	0.14	1	1	1	1	1	0.00	0.22	0.26	0.44	0.46	6.38	3.47	0.31
c2	em	1	1	1	1	1	0.03	0.02	0.06	0.10	0.04	5.25	6.62	0.36	1	1	1	1	1	0.00	0.15	0.18	0.38	0.47	6.18	3.73	0.57
c2	et3	1	1	1	1	1	0.04	0.06	0.11	0.13	0.03	5.37	6.43	0.30	1	1	1	1	1	0.00	0.28	0.22	0.39	0.50	6.39	3.80	0.49
c3	en	1	1	1	1	1	0.00	0.05	0.02	0.05	0.05	5.17	6.19	0.15	1	1	1	1	1	0.17	0.41	0.18	0.15	0.21	6.12	3.51	0.75
c3	em	1	1	1	1	1	0.01	0.08	0.00	0.02	0.04	5.15	6.45	0.39	1	1	1	1	1	0.16	0.37	0.25	0.15	0.20	6.13	3.82	0.92
c3	et3	1	1	1	1	1	0.04	0.05	0.06	0.05	0.07	5.27	6.25	0.30	1	1	1	1	1	0.22	0.35	0.15	0.15	0.27	6.14	3.94	0.98

Table 5.29: $p = 10$,$n = 100$ with Adapt1 by CV

cont		LS												RB										
x	y	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	MS	Ave(β)	Sd(β)
c0	en	1	1	1	1	1	0.06	0.05	0.07	0.03	0.02	5.23	3.78	0.23	1	1	1	1	1	0.13	0.09	0.07	0.09	0.06	5.44	3.19	0.20
c0	em	1	1	1	1	1	0.13	0.08	0.06	0.07	0.04	5.38	4.79	0.94	1	1	1	1	1	0.08	0.08	0.10	0.08	0.02	5.36	3.45	0.68
c0	et3	1	1	1	1	1	0.10	0.06	0.11	0.08	0.09	5.44	4.14	0.45	1	1	1	1	1	0.08	0.09	0.06	0.07	0.07	5.37	3.44	0.36
c1	en	1	1	1	1	1	0.01	0.19	0.04	0.06	0.15	5.45	7.68	0.10	1	1	1	1	1	0.09	0.09	0.09	0.14	0.09	5.50	7.14	0.50
c1	em	1	1	1	1	1	0.07	0.17	0.07	0.05	0.16	5.52	7.80	0.25	1	1	1	1	1	0.07	0.05	0.12	0.09	0.10	5.43	7.31	0.59
c1	et3	1	1	1	1	1	0.02	0.20	0.09	0.06	0.14	5.51	7.72	0.21	1	1	1	1	1	0.09	0.08	0.10	0.09	0.11	5.47	7.28	0.54
c2	en	1	1	1	1	1	0.01	0.22	0.23	0.14	0.02	5.62	7.56	0.10	1	1	1	1	1	0.00	0.07	0.13	0.09	0.13	5.42	7.14	0.36
c2	em	1	1	1	1	1	0.02	0.07	0.09	0.13	0.04	5.35	7.68	0.24	1	1	1	1	1	0.01	0.04	0.10	0.09	0.05	5.29	7.22	0.36
c2	et3	1	1	1	1	1	0.05	0.12	0.16	0.14	0.07	5.54	7.60	0.20	1	1	1	1	1	0.02	0.12	0.14	0.12	0.14	5.54	7.21	0.49
c3	en	1	1	1	1	1	0.01	0.02	0.01	0.07	0.02	5.13	7.57	0.10	1	1	1	1	1	0.13	0.33	0.18	0.27	0.20	6.11	6.76	0.56
c3	em	1	1	1	1	1	0.02	0.08	0.03	0.04	0.03	5.20	7.70	0.25	1	1	1	1	1	0.15	0.25	0.16	0.19	0.13	5.88	6.97	0.50
c3	et3	1	1	1	1	1	0.04	0.06	0.06	0.10	0.11	5.37	7.61	0.20	1	1	1	1	1	0.07	0.21	0.12	0.16	0.16	5.72	6.98	0.44

Table 5.30: $p = 10$,$n = 100$ with Adapt2 by CV

combinations of c_0, c_1, c_2 and three y outliers. RB has average model sizes closer to the true model size, 5, than LS and has smaller $RMSE(\beta)$. Under c_3, RB has smaller $RMSE(\beta)$ with slightly larger average model sizes. For example, Figure 5.9 shows the differences between LS and RB to estimate nonzero coefficient functions. The estimated coefficient curves under RB have closer to the true $\beta(t)$'s with narrower spreads than them under LS. In most cases, we can summarize that RB performs better than LS under the existence of outliers.

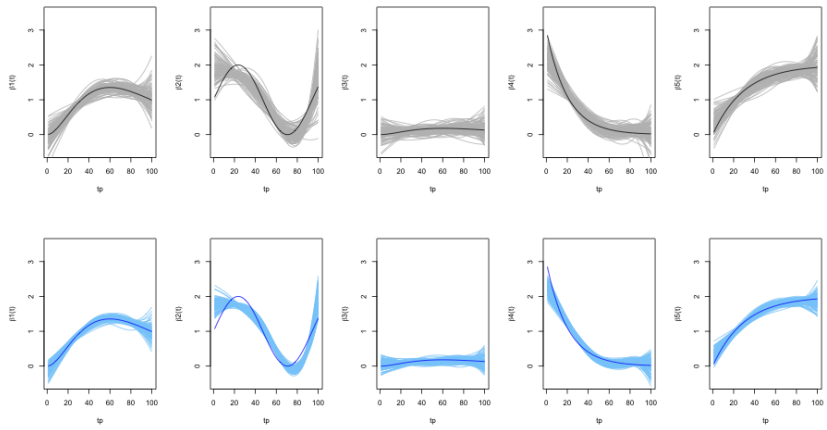

Figure 5.9: Estimated nonzero $\beta(t)$'s for c_0, em, Adapt 1 by CV, LS (grey), RB (blue)

5.5 Real Data Application: Weather Data

We apply the proposed rank-based method to analyze weather data in Matsui and Konishi [30] available in Chronological Scientific Tables 2005. The weather data includes monthly observed average temperatures (TEMP), average atmospheric pressure (PRESSURE), time of daylight (DAYLIGHT), average humidity (HUMIDITY), and annual total precipitation at 79 stations from 1971 to 2000 in Japan. We assume the annual total precipitation is a response variable depending four predictor functions, TEMP, PRESSURE, DAYLIGHT, and HUMIDITY in Figure 5.10 since these four predictors are trajectories over time.

Sawant [40] shows the curves of TEMP and PRESSURE for the 78th and 79th observations and the curves for the 1st, 2nd, and 3rd observations for the HUMIDITY are outliers. We see an outlier in the response on the box plot in Figure 5.11. The weather data set has outliers in both the predictor space and the response space. We approach to find the relation between

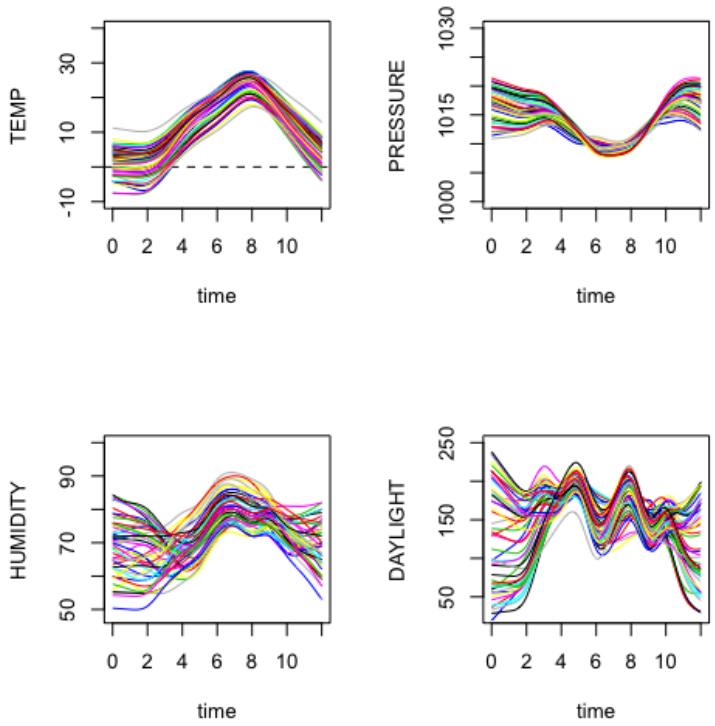

Figure 5.10: The predictors of Weather Data

predictor functions and the continuous discrete response using the multiple functional linear model in Equation (1.1).

$$y_i = \alpha + \sum_{j=1}^{4} \int_{\mathcal{T}} x_{ij}(t)\beta_j(t)dt + \varepsilon_i, \quad i = 1, \ldots, 79, \tag{5.6}$$

where y_i's are the annual total precipitations and $x_{ij}(t)$'s are TEMP, PRESSURE, HUMIDITY, and DAYLIGHT functions at $n = 79$ stations. We assume nonzero intercept exists.

First, we find the λ and φ which minimize the objective functions Q_{L_2,ℓ_2} in Equation (4.32) and Q_{aRB,ℓ_2} in Equation (5.1) by 10 fold cross-validation. We estimate the coefficient parameter functions for predictor functions using the optimal λ and φ for three methods.

We estimate the coefficient function curves for predictor functions. LS detects all four predictors as significant without adaptivity (Adapt0) and chooses PRESSURE as relevant with adaptivities. However, LAD detects one predictor, PRESSURE as significant with all adaptivities. RB selects three significant predictors, TEMP, HUMIDITY and DAYLIGHT with all

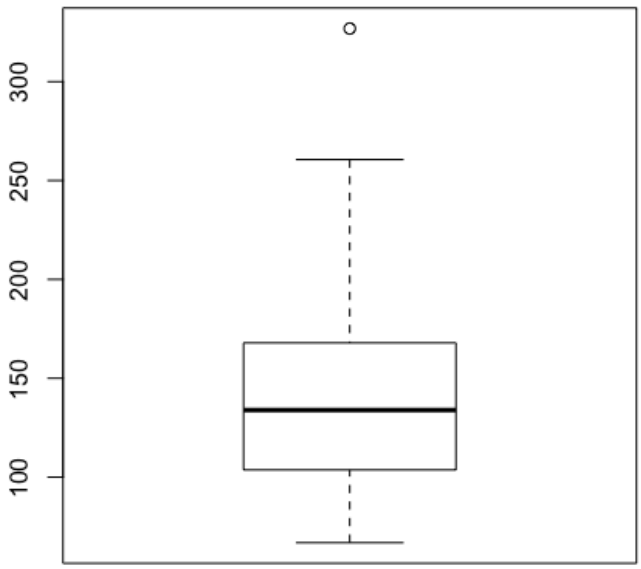

Figure 5.11: Boxplot of the response, annual average precipitation

adaptivities as in Table 5.31. LS and LAD have small values between 0 and 0.05 for PRESSURE with Adapt1 and Adapt2. With Adapt1 and Adapt2, RB has values close to the horizontal axis. The mean values of prediction error over 10-folds (CV-value) with Adapt2 are calculated as 133.42(LS), 134.70(LAD), and 169.35 (RB). In all loss functions, Adapt2 has slightly larger CV-values than Adapt1.

The estimated coefficient curves are in Figures 5.12, 5.13, and 5.14. In Adapt0, LS detects all predictors are significant and RB chooses three predictors except PRESSUE. TEMP and HUMIDITY have a positive effect to the response value since their estimated coefficient functions are positive over the range. It means that a positive amount of TEMP or HUMIDITY contribute positively to the response, the annual total precipitation. In TEMP, LS estimate gives increasing weight over time, but RB estimate has a constant weight over time. In HUMIDITY, both LS and RB have similar decreasing weight over time. DAYLIGHT estimate functions for LS and RB has positive and negative values at the same time. Under LS method, Spring time has a positive contribution of Daylight to the response and other seasons have negative

		TEMP	PRESSURE	HUMIDITY	DAYLIGHT	CV
Adapt0	LS	✓	✓	✓	✓	90.29
	LAD	--	✓	--	--	134.23
	RB	✓	--	✓	✓	214.52
Adapt1	LS	--	✓	--	--	133.40
	LAD	--	✓	--	--	134.68
	RB	✓	--	✓	✓	160.91
Adapt2	LS	--	✓	--	--	133.42
	LAD	--	✓	--	--	134.70
	RB	✓	--	✓	✓	169.35

Table 5.31: Relevant Predictors for Weather Data

contribution. Under RB method, Daylight is negative from January to October and positive after October. The estimates of PRESSURE are close or identical to zero compared to other estimated coefficient functions in all three loss methods.

Adapt1 and Adapt2 choose only one coefficient for PRESSURE for LS and LAD which is close to zero. However, RB chooses the same three predictors as significant. Similarly to Adapt0, TEMP and HUMIDITY have positive values and DAYLIGHT has negative values over time.

5.6 Conclusion

We establish the rank-based method with preserving group structures by using the group ℓ_2 penalty. By using the group ℓ_2 penalty, we can obtain only between-group sparsity to express a functional coefficient precisely by taking as many as possible nonzero coefficients for all basis functions. The rank-based loss function with the weight b_{ij} to control observations with high leverage values in predictor space. Also, the rank-based loss function takes care of non-normal errors in the response space. The rank-based loss functions performs best among all loss functions and it endures all kinds of outliers to estimate the coefficient functions and the model size close to the true ones, especially in the simulation result with $p = 10$. Also, the regularized rank-based method for functional model achieves the oracle property with adaptivities in Appendix. The proposed method for functional multiple linear model can control the smoothness of the estimated coefficient functions with the smoothness penalty φ. Compared to LS and LAD, RB detects a meaningful and size-able set of predictors in a real data example.

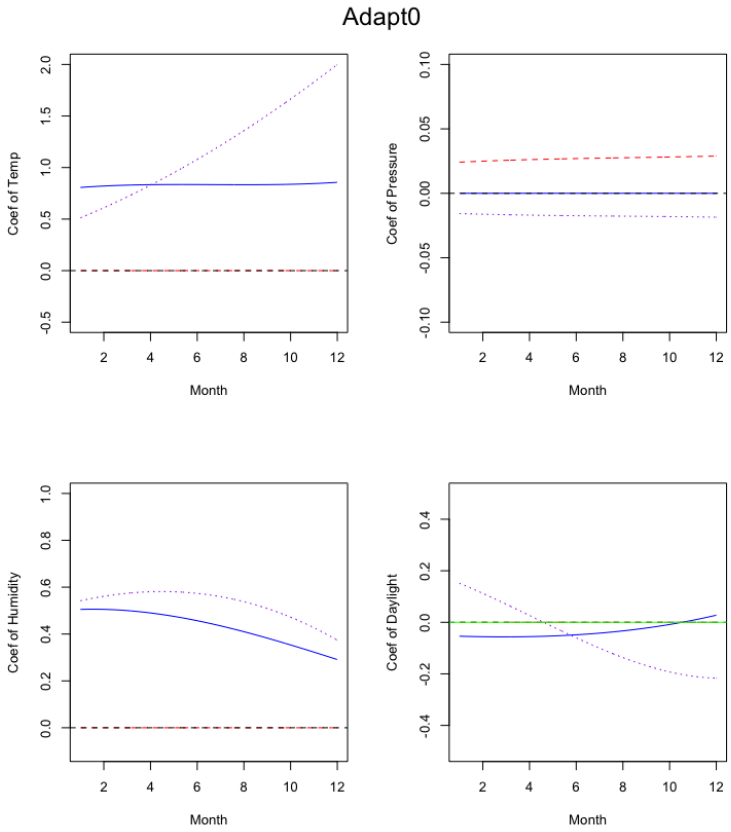

Figure 5.12: Estimated coefficients for Weather data of LS(purple), LAD(red), and RB(blue) with Adapt0

However, it is challenging to find the optimal tuning parameters. It is computationally expensive and there are numerous local minima of λ with CV and SIC. SIC or other criteria except CV might depend on the combination of the number of basis for function, the sample size, and errors in the response.

One extension of the proposed method can be to establish a proper relation between the number of basis, the sample size, and errors to find the optimal tuning parameter for rank-based loss function using SIC, BIC, AIC, GACV, or GCV.

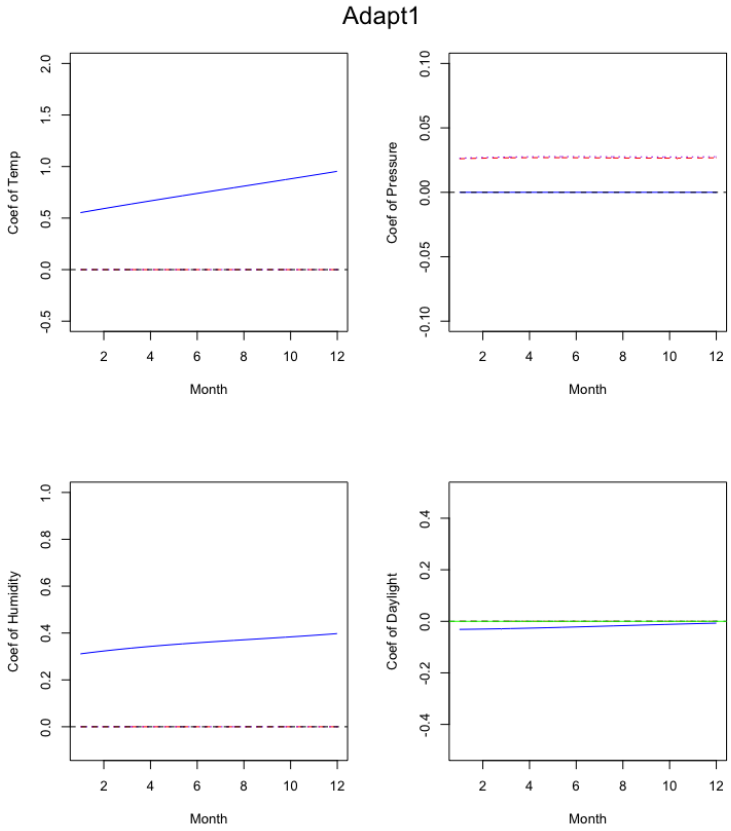

Figure 5.13: Estimated coefficients for Weather data of LS(purple), LAD(red), and RB(blue) with Adapt1

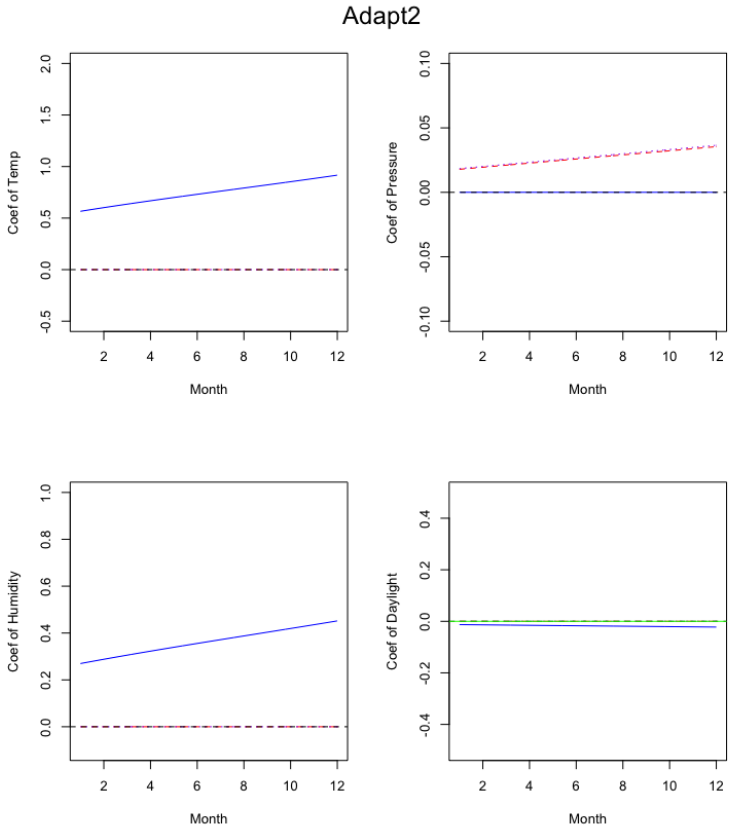

Figure 5.14: Estimated coefficients for Weather data of LS(purple), LAD(red), and RB(blue) with Adapt2

References

[1] A. Abebe and H. F. Bindele. Robust signed-rank variable selection in linear regression. In *Robust Rank-Based and Nonparametric Methods*, pages 25–45. Springer, 2016.

[2] Z. Bai, C. R. Rao, and Y. Yin. Least absolute deviations analysis of variance. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 166–177, 1990.

[3] J. L. Bali, G. Boente, D. E. Tyler, J.-L. Wang, et al. Robust functional principal components: A projection-pursuit approach. *The Annals of Statistics*, 39(6):2852–2882, 2011.

[4] S. Bang and M. Jhun. Simultaneous estimation and factor selection in quantile regression via adaptive sup-norm regularization. *Computational Statistics & Data Analysis*, 56(4):813–826, 2012.

[5] H. F. Bindele, A. Abebe, and P. Zeng. Robust estimation and selection for single-index regression model. *Journal of Statistical Computation and Simulation*, 89(8):1376–1393, 2019.

[6] D. Birkes and Y. Dodge. *Alternative methods of regression*. Wiley Online Library, 1993.

[7] M. Denhere and N. Billor. Robust principal component functional logistic regression. *Communications in Statistics-Simulation and Computation*, 45(1):264–281, 2016.

[8] M. Denhere and H. F. Bindele. Rank estimation for the functional linear model. *Journal of Applied Statistics*, 43(10):1928–1944, 2016.

[9] M. Escabias, A. Aguilera, and M. Valderrama. Principal component estimation of functional logistic regression: discussion of two different approaches. *Journal of Nonparametric Statistics*, 16(3-4):365–384, 2004.

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

[11] R. Fraiman and G. Muniz. Trimmed means for functional data. *Test*, 10(2):419–440, 2001.

[12] J. Gertheiss, A. Maity, and A.-M. Staicu. Variable selection in generalized functional linear models. *Stat*, 2(1):86–101, 2013.

[13] C. Gutenbrunner and J. Jurecková. Regression rank scores and regression quantiles. *The Annals of Statistics*, pages 305–330, 1992.

[14] T. P. Hettmansperger and J. W. McKean. *Robust nonparametric statistical methods*. Arnold, 1998.

[15] Z. Hong and H. Lian. Inference of genetic networks from time course expression data using functional regression with lasso penalty. *Communications in Statistics-Theory and Methods*, 40(10):1768–1779, 2011.

[16] L. Horváth and P. Kokoszka. *Inference for functional data with applications*, volume 200. Springer Science & Business Media, 2012.

[17] L. Horváth and G. Rice. An introduction to functional data analysis and a principal component approach for testing the equality of mean curves. *Revista matemática complutense*, 28(3):505–548, 2015.

[18] L. A. Jaeckel. Estimating regression coefficients by minimizing the dispersion of the residuals. *The Annals of Mathematical Statistics*, pages 1449–1458, 1972.

[19] G. M. James. Generalized linear models with functional predictors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):411–432, 2002.

[20] B. A. Johnson and L. Peng. Rank-based variable selection. *Journal of Nonparametric Statistics*, 20(3):241–252, 2008.

[21] J. Jureckova. Nonparametric estimate of regression coefficients. *The Annals of Mathematical Statistics*, pages 1328–1338, 1971.

[22] K. Kato. Group lasso for high dimensional sparse quantile regression models. *arXiv preprint arXiv:1103.1458*, 2011.

[23] H.-J. Kim, E. Ollila, and V. Koivunen. New robust lasso method based on ranks. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 699–703. IEEE, 2015.

[24] J. Kloke and J. W. McKean. *Nonparametric statistical methods using R*. Chapman and Hall/CRC, 2014.

[25] R. Koenker. Quantile regression for longitudinal data. *Journal of Multivariate Analysis*, 91(1):74–89, 2004.

[26] R. Koenker and G. Bassett. Tests of linear hypotheses and l" 1 estimation. *Econometrica: Journal of the Econometric Society*, pages 1577–1583, 1982.

[27] R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[28] R. Koenker, V. Chernozhukov, X. He, and L. Peng. *Handbook of Quantile Regression*. CRC press, 2017.

[29] K. Lilly. *Robust variable selection methods for grouped data*. PhD thesis, Auburn University, 2015.

[30] H. Matsui and S. Konishi. Variable selection for functional regression models via the l1 regularization. *Computational Statistics & Data Analysis*, 55(12):3304–3310, 2011.

[31] L. Meier, S. Van de Geer, P. Bühlmann, et al. High-dimensional additive modeling. *The Annals of Statistics*, 37(6B):3779–3821, 2009.

[32] G.-v. M. Miakonkana, B. M. Nguelifack, and A. Abebe. Rank-based group variable selection. *Journal of Nonparametric Statistics*, 28(3):550–562, 2016.

[33] N. Mingotti, L. Rodríguez, R. Elvira, and J. Romo Urroz. Lasso variable selection in functional regression. *Statistics and Econometrics Series 13, Working paper*, pages 13–14, 2013.

[34] H.-G. Müller, U. Stadtmüller, et al. Generalized functional linear models. *the Annals of Statistics*, 33(2):774–805, 2005.

[35] J. D. Naranjo and T. Hettmansperger. Bounded influence rank regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):209–220, 1994.

[36] A. B. Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007.

[37] J. Pannu and N. Billor. Robust group-lasso for functional regression model. *Communications in Statistics-Simulation and Computation*, 46(5):3356–3374, 2017.

[38] J. O. Ramsay and B. W. Silverman. *Functional data analysis*. Springer, 2005.

[39] S. Rosset and J. Zhu. Discussion of" least angle regression" by efron et al. *arXiv preprint math/0406470*, 2004.

[40] P. Sawant, N. Billor, and H. Shin. Functional outlier detection with robust functional principal component analysis. *Computational Statistics*, 27(1):83–102, 2012.

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

[42] H. Wang and C. Leng. A note on adaptive group lasso. *Computational statistics & data analysis*, 52(12):5277–5286, 2008.

[43] H. Wang, G. Li, and G. Jiang. Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355, 2007.

[44] L. Wang and R. Li. Weighted wilcoxon-type smoothly clipped absolute deviation method. *Biometrics*, 65(2):564–571, 2009.

[45] Y. Wu and Y. Liu. Variable selection in quantile regression. *Statistica Sinica*, 19(2):801, 2009.

[46] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[47] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Appendices

Appendix A

Oracle Property in RB Loss with Adaptive Group ℓ_2 Penalty

A.1 Oracle Property on Discrete Multiple Linear Model

We consider the estimation consistency, the variable selection consistency and the oracle property for the rank-based group variable selection with ℓ_2 penalty.

We show that the group ℓ_2 penalized rank-based variable selection estimator has the oracle property under some regularity conditions. In this section, we follow the definition and notation as Miakonkana et al.[32] and Wang and Li [44]. We assume that only the first $k_0 \leq K$ groups are significant, that is, $||\boldsymbol{\beta}_k||_2 \neq 0$ for $k \leq k_0$ and $||\boldsymbol{\beta}_k||_2 = 0$ for $k > k_0$. Denote $\boldsymbol{\beta}_0$ the true parameter, $\boldsymbol{\beta}_a$ the vector containing all relevant groups and $\boldsymbol{\beta}_b$ the vector of all irrelevant groups. Let $\hat{\boldsymbol{\beta}}_a$ and $\hat{\boldsymbol{\beta}}_b$ be their corresponding penalized rank-based estimator.

The following regularity conditions will be assumed.

C1. The errors ϵ_i are iid with a density function f that is absolute continuous and has a finite fisher informations. That is,

$$I(f) = \int_{-\infty}^{\infty} \Big[\frac{f'(e)}{f(e)}\Big]^2 f(e)de < \infty$$

C2. The matrices $\mathbf{X}$ and $\mathbf{WX}$ satisfy the Huber's condition.

C3. $n^{-1}\mathbf{X}'\mathbf{WX} \xrightarrow{\text{P}} \mathbf{C}$, and $n^{-1}\mathbf{X}'\mathbf{X} \xrightarrow{\text{P}} \boldsymbol{\Sigma}$ are positive definite matrices.

given by

$$\mathbf{C} = \frac{1}{2}\int\int(\mathbf{x}_2 - \mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1)'b(\mathbf{x}_1, \mathbf{x}_2)dM(\mathbf{x}_2)dM(\mathbf{x}_1)$$
$$\mathbf{V} = \int\{\int(\mathbf{x}_2 - \mathbf{x}_1)b(\mathbf{x}_1, \mathbf{x}_2)dM(\mathbf{x}_2)\}\{\int(\mathbf{x}_2 - \mathbf{x}_1)b(\mathbf{x}_1, \mathbf{x}_2)dM(\mathbf{x}_2)\}'dM(\mathbf{x}_1)$$
$$\mathbf{\Sigma} = \frac{1}{2}\int\int(\mathbf{x}_2 - \mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1)'dM(\mathbf{x}_2)dM(\mathbf{x}_1)$$

and $M(\mathbf{x})$ denotes the CDF of $\mathbf{x}$, $\mathbf{X}$ is a matrix whose rows are $\mathbf{x}_i$, and the entries ω_{ij} of the matrix $\mathbf{W}$ are defined like in Naranjo and Hettmansperger (1994)[35], defined by

$$\omega_{ij} = \begin{cases} n^{-1}b_{ij} & \text{if} \quad i \neq j \\ n^{-1}\sum_{k\neq i} b_{ij} & \text{if} \quad i = j \end{cases} \tag{A.1}$$

We derive conditions for model selection and estimation consistency when when the sample size n increases.

Following the notation in Wang and Leng (2008)[42] define

$$a_n = \max\{\lambda_{kj} : 1 \leq j \leq k; k \leq k_0\} \text{ and } b_n = \min\{\lambda_{kj} : 1 \leq j \leq k; k > k_0\},$$

and $H(\mathbf{x}, y)$ be the joint distribution between the covariate $\mathbf{x}$ and the response variable y.

Theorem A.1. *Let* $(y_1, \mathbf{x}_1), \cdots, (y_n, \mathbf{x}_n)$ *be independent and identically distributed from* $H(\mathbf{x}, y)$. *Assume the regularity conditions* C1–C3.

a. If $\sqrt{n}a_n \xrightarrow{P} 0$ *then* $||\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_0||_2 = O_p(n^{-1/2})$

b. If $\sqrt{n}a_n \xrightarrow{P} 0$ *and* $\sqrt{n}b_n \xrightarrow{P} \infty$ *then* $\hat{\boldsymbol{\beta}}_b \xrightarrow{P} 0$

c. Under local shrinking contamination, $H_n^*(\mathbf{x}, y)$, $\sqrt{n}(\hat{\boldsymbol{\beta}}_a - \boldsymbol{\beta}_a) \xrightarrow{D} N(\eta, \tau^2 C_{11}^{-1} V_{11} C_{11}^{-1})$

To prove Theorem A.1, we define the following expressions defined in Wang and Li [44] with the group ℓ_2 penalty.

$$
\begin{aligned}
Q_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i<j} b_{ij}|\epsilon_i - \epsilon_j| + n\sum_{k=1}^{K}\Big(\sum_{j=1}^{p_k}(\lambda_{kj}\theta_{kj})^2\Big)^{1/2} \\
D_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i<j} b_{ij}|\epsilon_i - \epsilon_j| \\
S_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i<j} b_{ij}(\mathbf{x}_i - \mathbf{x}_j)sgn((y_i - y_j) - (\mathbf{x}_i - \mathbf{x}_j)'\boldsymbol{\theta}) \\
A_n(\boldsymbol{\theta}) &= (2\sqrt{3\tau})^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)'\mathbf{X}'\mathbf{W}\mathbf{X}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) - (\boldsymbol{\theta} - \boldsymbol{\theta}_0)'S_n(\boldsymbol{\theta}_0) + D_n(\boldsymbol{\theta}_0)
\end{aligned}
$$

Every above expression is identical to the one in Wang and Li [44] except the group ℓ_2 penalty. We can borrow the result of the following lemma.

Lemma A.1. *Under assumptions C1–C3,*

i. for all $\epsilon > 0$ *and* $c > 0$*,*

$$
[\sup_{\sqrt{n}||\boldsymbol{\theta}-\boldsymbol{\theta}_0||\leq c} |D_n(\boldsymbol{\theta}) - A_n(\boldsymbol{\theta})| \geq \epsilon] \xrightarrow{P} 0
$$

under either H *or* H_n^**,*

ii. $n^{-1/2}S_n(\boldsymbol{\theta}_0) \xrightarrow{D} N(0, \mathbf{V}/3)$ *under* H*,*

iii. $n^{-1/2}S_n(\boldsymbol{\theta}_0) \xrightarrow{D} N(\eta, \mathbf{V}/3)$ *under* H_n^**.*

We follow the same logic to Miakonkana et al. for the proof of Theorem A.1 with the group adaptive ℓ_2 penalty instead of the group and element-wise adaptive ℓ_1 penalty.

Proof. To prove part (a), it is sufficient to show that $\forall\epsilon > 0$, there exists a large constant C such that

$$
P\Big(\inf_{||u||=C} Q_n(\boldsymbol{\theta}_0 + n^{-1/2}\mathbf{u}) > Q_n(\boldsymbol{\theta}_0)\Big) \geq 1 - \epsilon
$$

where $\mathbf{u}$ is a vector of dimension p. Since $Q_n(\boldsymbol{\theta})$ is convex in $\boldsymbol{\theta}$, this implies that with probability at least $1 - \epsilon$ the penalized estimator lies in the ball $\{\boldsymbol{\theta}_0 + n^{-1/2}\mathbf{u} : ||\mathbf{u}|| \leq C\}$. Let $G_n(\mathbf{u}) = Q_n(\boldsymbol{\theta}_0 + n^{-1/2}\mathbf{u}) - Q_n(\boldsymbol{\theta}_0)$. Denote by u_{kj} the component of $\mathbf{u}$ corresponding to θ_{kj}.

By Lemma A.1,

$$
\begin{aligned}
G_n(\mathbf{u}) &= (2\sqrt{3})^{-1}\mathbf{u}'[n^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}]\mathbf{u} - \mathbf{u}'n^{-1/2}S_n(\boldsymbol{\theta}_0) \\
&\quad + n\sum_{k=1}^{K}\Big[\Big(\sum_{j=1}^{p_k}(\lambda_{kj}(\theta_{kj}+n^{-1/2}u_{kj}))^2\Big)^{1/2} - \Big(\sum_{j=1}^{p_k}(\lambda_{kj}\theta_{kj})^2\Big)^{1/2}\Big] + o_p(1) \\
&\geq (2\sqrt{3})^{-1}\mathbf{u}'[n^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}]\mathbf{u} - \mathbf{u}'n^{-1/2}S_n(\boldsymbol{\theta}_0) - \sqrt{n}\sum_{k=1}^{k_0}\Big(\sum_{j=1}^{p_k}(\lambda_{kj}u_{kj})^2\Big)^{1/2} + o_p(1) \\
&= (2\sqrt{3})^{-1}\mathbf{u}'[n^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}]\mathbf{u} - \mathbf{u}'O_p(1) - \sqrt{n}\sum_{k=1}^{k_0}\Big(\sum_{j=1}^{p_k}(\lambda_{kj}u_{kj})^2\Big)^{1/2} + o_p(1) \\
&\geq (2\sqrt{3})^{-1}\mathbf{u}'[n^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}]\mathbf{u} - \mathbf{u}'O_p(1) - k_0\sqrt{n}a_n(||\mathbf{u}||_2) + o_p(1).
\end{aligned}
$$

Note that $n^{-1}\mathbf{X}'\mathbf{W}\mathbf{X} \xrightarrow{\text{P}} \mathbf{C}$, a positive definite matrix, and $\sqrt{n}a_n \xrightarrow{\text{P}} 0$. Therefire, for n sufficiently large, the first term on the right hand side of the ineuqality above dominates. $G_n(\mathbf{u})$ can be made positive when the size of ball C is chosen to be sufficiently large. We now prove part (b). Suppose that $\widehat{\boldsymbol{\theta}}_b \neq 0, \forall n \in \mathbb{N}$. Let k be such that $k_0 < k < K$ and $\widehat{\theta}_{kj} \neq 0$ for some j such that $1 \leq j \leq p_k$. Since $Q_n(\boldsymbol{\theta})$ is differentiable at any point, except the origin, $\widehat{\theta}_{kj}$ must be solution of the equation

$$
0 = n^{-3/2}\sum_{i<j} b_{ij}(\mathbf{x}_{ik} - \mathbf{x}_{jk})sgn(y_i - y_j) - (\mathbf{x}_i - \mathbf{x}_j)'\boldsymbol{\theta}) + \sqrt{n}\lambda_{kj}sgn(\theta_{kj}).
$$

Now, by the consistency of $\widehat{\boldsymbol{\theta}_n}$ and part (ii.) of lemma A.1, the first term of the right hand side of the equation above is $O_p(1)$. In addition, $\sqrt{n}b_n \xrightarrow{P} \infty$ implies that $\sqrt{n}\lambda_{kj} \xrightarrow{P} \infty$. So the equation does not hold for large values of n, as we assume that $\widehat{\theta}_{kj} \neq 0$. Therefore, $\widehat{\boldsymbol{\theta}}_b \xrightarrow{P} 0$.

The proof of part (c) is identical to the proof of Theorem 2 given in the Web Appendix of Wang and Li (2009)[44], and will therefore be omitted here.

□

A.2 Oracle Property on Functional Linear Model

We convert the functional linear model in Equation (2.1) to the discretized model in Equation (5.1) considering the functional group adaptive penalty. Similarly, we can see the oracle property of the rank-based estimates with the adaptive group ℓ_2 penalty for functional linear model.