Robust Bayesian Methods for Semi-parametric Models

by

Wei Huang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Department of Mathematics and Statistics

Auburn, Alabama
August 8, 2020

Keywords: Wilcoxon rank-based estimate, Bayesian, single-index model, measurement error, varying coefficient model, robust

Approved by

Asheber Abebe, Chair, Professor of Mathematics and Statistics
Mark Carpenter, Professor of Mathematics and Statistics
Peng Zeng, Associate Professor of Mathematics and Statistics
Guanqun Cao, Associate Professor of Mathematics and Statistics

Abstract

Nonparametric rank-based approaches in many situations provide more flexible modeling specifications and robustness when the distribution of data differs from the assumed distribution. This dissertation is mainly concerned with two robust Bayesian methods using the ideas of rank-based approaches and least absolute deviations estimate.

We use simple linear model to propose our Bayesian Wilcoxon rank-based estimate, and applied this estimate in linear model with measurement error, single-index model and varying coefficient model. We use two classical real data examples to show that our Bayesian Wilcoxon rank-based estimate is more efficient than normal rank-based estimate. Some extensive simulation studies are proceed to demonstrate that Bayesian Wilcoxon rank-based estimate successfully brings robustness to the results, and it also inherits several advantages of Bayesian inference.

Another robust Bayesian method base on the idea of least absolute deviations (LAD) estimate is also proposed in single-index model, and it is applied on single-index varying coefficient model as well. The simulation studies show that the Bayesian LAD method provides similar result to Bayesian Wilcoxon rank-based estimate with much less computation cost, and it does not lose much efficiency. The procedure of Bayesian LAD method is much easier to achieve which makes it more feasible when dealing with complicated models.

Acknowledgments

This dissertation would not have been possible without the support and help of many people.

Foremost, I would like to take this opportunity to express my sincere gratitude to my advise, Dr. Asheber Abebe for his continuous support of my Ph.D. studies. I appreciate his invaluable guidance and assistance, as well as his patience, and immense knowledge. I have learned a lot from his deep insight into statistics. I could not imagine having a better advisor and friend in my life.

I wish to thank the rest of my dissertation committee, Dr. Peng Zeng, Dr. Guanqun Cao and Dr. David Mark Carpenter for their assistance and comments. This research would not have been successful without them. I am also grateful to other faculty members who taught me or helped me in the mathematics and statistics department. Thank you for your excellent lectures and guidance.

I appreciate the friendship with my classmates and friends, even if many of them left Auburn a few years ago. My time at Auburn was enjoyable because of them.

Finally, I would like to dedicate this thesis to my family, especially my wife, for her constant encouragement and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Nonparametric rank-based approaches in many situations provide more flexible modeling specifications, can be much more efficient compared to traditional parametric methods when the distribution of data differs from the assumed distribution, and when the assumed model is correct, do not lose much efficiency. Meanwhile, the Bayesian approach to inference is appealing in incorporating prior information by Bayes ' theorem into the inference machinery, resulting in a unifying, constructive methodology of inference.

Historically, Bayesian modeling and non-parametric notions have been considered incompatible. In order to clarify the difficulty, note that the parameter space of a nonparametric model consists of one or more unknown functions. One assumes that unknown functions belong to some appropriate function classes. Under this setup, the Bayesian modeling, assuming all unknowns are random, requires the unknown functions to be random and therefore requires priors to be used over function spaces. In contrast to the parametric case in which the dimension of the space parameter is finite, Bayesian nonparametric modeling requires an infinite dimensional parameter and therefore a prior parameter over an infinite dimension.

The mainstream class of approaches in Bayesian nonparametric based on the Dirichlet process priors, for which a considerable amount of research has been carried out over the past forty years. The Dirichlet process, a random probability measure for a probability measurement space, was the first prior on a function space, specifically the distribution function space. It was formally developed by Ferguson (1973, 1974) and is now commonly referred to as Bayesian nonparametrics after Ferguson's work has grown rapidly. In the above-mentioned Bayesian nonparametric modeling, priors are assumed over function spaces, but the usual method of calculating the posterior distribution, i.e. "the posterior is proportional to the likelihood times the prior", is not applicable here, since there is no likelihood. As a result, the posterior distribution is obtained in a different way. Zhan (2009) discussed a Bayesian approach to applying non-parametric rank-based methodology to linear models instead of postulating a prior to a function space. They only assume a prior distribution for the parameter(s)θ of interest in the unknown function. In the mean time, they summarize the information in a sample of data x via the distribution of a certain quantity, say $T(X, \theta)$, and use that distribution as a pseudo-likelihood. After applying the Bayes theorem directly, they can obtain the complete posterior distribution of $T(X, \theta)$. In this formulation, no specific form of the underlying distribution of data is assumed, but we specify certain parameters of interest for the unknown function and assign priors to these parameters.

One simple way to apply Bayesian rank-based estimators is to calculate the complete posterior distribution for the parameter(s) θ of interest under the frame created by Zhan (2009) in linear model. The disadvantage of this methodology is that, we need to figure out the complete posterior distribution(s) for the parameter(s) when the specification of the model changes. To circumvent this practical problem, we present a new method to provide a simpler way to obtain Bayesian rank-based estimators in several models. Consider a general model with additive errors and parameterized by θ given as $y_i = f(\boldsymbol{x}_i, \theta) + \varepsilon_i$, for $i = 1, \ldots, n$. Our specification requires a formulation of the problem in the Mann-Whitney-Wilcoxon rank test framework where we need to create paired differences of the errors $\varepsilon_i - \varepsilon_j$ for $j > i$. This means we will be dealing with $y_i - y_j$ for all $j > i$ as the response, and correspondingly $f(\boldsymbol{x}_i, \theta) - f(\boldsymbol{x}_j, \theta)$ as the "model". In addition to the added computational burden of having to deal with $\binom{n}{2}$ data points, we also note that the responses are no longer independent of each other even if ε are independent and identically distributed.

From the well-established theory of multivariate linear regression (Seber and Lee, 2003), we know that while the correlation among response variables will not affect model parameter estimates' unbiasedness and consistency, it will affect the variance of the estimated parameters. We could figure out the effect of this dependence under the Bayesian framework for each of the different model specifications; however, that goes against our motivation of creating a robust Bayesian estimation framework that can be easily applied on different models. Using a working correlation structure is another reasonable way to fix the problem. Since the working correlation structures for many models are known, we can ignore the covariance first, then we apply the working correlation structure to correct the variance of the estimated parameters.

Following a similar approach used for obtaining the aforementioned new Bayesian Wilcoxon rank-based estimator, we can also calculate the least absolute deviation (LAD) estimator under Bayesian framework for any additive model specification. The Bayesian LAD estimator, however, does not require the pairing of the data, hence avoiding correlated response issue that we faced in the case of Bayesian Wilcoxon estimation. At the same time, this is also a new robust Bayesian estimation approach with much lower computational cost than the Bayesian Wilcoxon rank-based estimator. This gain in computation time comes at a cost of loss in efficiency (Hettmansperger and McKean, 2011).

1.2 Motivation

Our research was initially motivated by ideas raised in two papers. The first is Zhan (2009), which has been discussed in Section 1.1, and the second is Jureckova et al. (2016) which will be discussed in Section 3.1. We started out with the goal of providing a Bayesian rank-based estimator in linear model with measurement error; that is, the design variables are measured with error. We quickly noticed that it was very difficult to derive the complete posterior distribution of the parameters in linear model with measurement error through a combination of ideas given in these two papers. On the other hand, with proper specification of the likelihood function in software for Bayesian analysis such as JAGS (Plummer, 2003), we should be able to obtain robust estimates measurement error issue in linear model easily. We were interested in developing a robust Bayesian approach which can take advantage of the machinery provided by existing software like JAGS and can be easily used for a variety of model specifications.

Following a successful application of the proposed method in the linear model with measurement error, we extended the approach to other models. One such model we considered was the single-index model (SIM) proposed by Ichimura (1993). There is some existing research for robust rank-based nonparametric estimation in single-index model (Bindele et al., 2018), confidence intervals (Bindele et al., 2018), and variable selection (Bindele et al., 2019). The approach used in these papers shows that the SIM is treated quite differently from the linear model. However, the method we develop in this dissertation is flexible enough to be used for SIM estimation. Theoretically, our method should be able to proceed as long as we can express an appropriate likelihood and model specification in software like JAGS. An interesting extension would be the SIM under a measurement error scenario. We show that the classical Bayesian approach based on a Gaussian likelihood could be used for estimation. We can also use the Bayesian LAD method. However, the Bayesian Wilcoxon rank-based method presents a practical challenge due to the lack of identifiability. This problem will be discuss in Section 4.4.

Since the SIM is the intercept case of the single-index varying coefficient model (SIVCM), we were interested in whether our robust Bayesian approach could be applied in the more complex SIVCM setup. Again, there is some existing research on rank-based estimation of the SIVCM (Sun et al., 2019, Abebe et al., 2020), but none of them use our Bayesian approach. We explore the possibility of extending the proposed Bayesian computation approach to obtain robust nonparametric estimates of SIVCM parameters as well as the coefficient functions.

1.3 Contribution

The main contribution of this dissertation is providing a Bayesian Wilcoxon rank-based estimation procedure that can be used for a variety of models with additive random error components.

In Chapter 2, we consider the linear model to develop the Bayesian Wilcoxon rank-based method. We explain the idea of using Laplace distribution as the likelihood on paired data to obtain the Bayesian Wilcoxon rank-based estimate. With two real data examples, we demonstrate that the resulting estimator is as robust as general Wilcoxon rank-based estimator compared to the Bayesian Least Squares (BLS) estimator when the data contain several outliers. The cross-validation result showed that the Bayesian Wilcoxon rank-based estimator provides smaller prediction errors in a majority of the cases considered. Since the pairwise differenced samples are no longer independent of each other, the uncertainty of the estimated coefficients that is obtained by ignoring the covariance structure will be possibly biased. We discuss a possible means to address the problem using sandwich variance.

In Chapter 3, the linear model with measurement error scenario is introduced. We used our Bayesian Wilcoxon rank-based method which is proposed in Chapter 2 to show that this method can also be easily adapted for use on different models other than the regular linear model, including the linear model with measurement error in the covariates. Our simulation studies demonstrate that our approach that utilized the Bayesian framework handles measurement error successfully while maintaining the robustness simultaneously. Compared to traditional approaches of dealing with measurement error, our method is much more flexible and efficient when the model error follows a heavy tailed distribution or the data contain outliers.

The single index model (SIM) is introduced in Chapter 4. We applied the approach which is proposed in Chapter 2 on SIM. Moreover, to avoid issue of dependence effects on biasing the standard errors of parameter estimates, we propose a new least absolute deviation (LAD) method under Bayesian framework which is also robust with some trade off in efficiency. Simulations illustrate that both the Bayesian Wilcoxon rank-based method and the least absolute deviation method provide similar robustness. We applied these two methods on a real data example. Again, we used a working correlation structure to correct the bias on the standard errors caused by covariance when using the Bayesian Wilcoxon rank-based estimate.

In Chapter 5, we applied our robust Bayesian methods on varying coefficient (VC) model, and discussed possible applications on SIVCM. A simulation study was provided to show that our methods outperformed the BLS method for VC model when there are outliers in the data or the error distribution is non-normal. We demonstrated an application on a real data example where we can see that the Bayesian LAD approach we proposed is robust and efficient in comparison to BLS method.

Chapter 2

Bayesian Wilcoxon Estimate in Linear Model

2.1 Introduction to Bayesian Wilcoxon Rank-based Estimate in Linear Model

Without loss of generality, we consider the simplest linear regression model

$$Y_i = \alpha + \beta x_i + \varepsilon_i$$

The least squares estimator can be obtained by minimizing the sum of squared residuals

$$\widehat{\beta}_{LS} = \text{Argmin } \|\boldsymbol{\varepsilon}\|_2$$

where $\|\boldsymbol{\varepsilon}\|_2$ is the square of the Euclidean norm on $\mathbb{R}^n$ given by

$$\|\boldsymbol{\varepsilon}\|_2 = \sum_{i=1}^{n} \varepsilon_i^2 \ .$$

This is equivalent to maximizing the Gaussian likelihood given by

$$Y_i \sim N\,(\beta_0 + \beta_1 x_i, \sigma^2).$$

This will be the likelihood specification in classical Bayesian estimation.

Since we do not want to impose strong parametric assumptions on the underlying population as the above classical Bayesian approach, the less restrictive assumption under which a better procedure is possible seems to be absolute continuity of the error distribution. Among classical rank-based statistics, the Wilcoxon rank statistic is one that makes this assumption, in addition to finite Fisher information. In this case, the rank-based Wilcoxon estimator of β is found as (Jaeckel, 1972)

$$\widehat{\beta}_R = \text{Argmin } \|\boldsymbol{\varepsilon}\|_R$$

where

$$\|\boldsymbol{\varepsilon}\|_R = 4\sum_{i=1}^{n}\left(R\left(\varepsilon_i\right) - \frac{n+1}{2}\right)\varepsilon_i \tag{2.1}$$

where $R\left(\varepsilon_i\right) = \sum_{j=1}^{n} \mathbf{1}(\varepsilon_j \leq \varepsilon_i)$, where $\mathbf{1}(A)$ is the indicator of the set A, is the rank of ε_i among the other residuals.

It can easily be shown that $\|\boldsymbol{\varepsilon}\|_2$ is a norm as well as convex and non-negative as a function of β. Jaeckel (1972) showed that $\|\boldsymbol{\varepsilon}\|_R$ is also convex and non-negative as a function of β. However, it is easy to show that it is not a norm. It satisfies all the properties of a norm except $\|\boldsymbol{\varepsilon}\|_R = 0$ does not imply $\boldsymbol{\varepsilon} = \mathbf{0}$. Instead, it is a pseudo-norm (McKean and Schrader, 1980) and this will be apparent in our development below. This is also the reason while it can be used to estimate the slope β of the linear model, it cannot provide an estimate of the intercept α. We can estimate the intercept following the estimation of the slope (Hettmansperger and McKean, 2011; Sievers and Abebe, 2004) but for this dissertation we will focus solely on the estimation of the slope parameter.

We will demonstrate below that $\|\boldsymbol{\varepsilon}\|_R$ can be rewritten as

$$\|\boldsymbol{\varepsilon}\|_R = \sum_{i=1}^{n}\sum_{j=1}^{n} \left|\varepsilon_i - \varepsilon_j\right|, \quad \boldsymbol{\varepsilon} \in R^n\ . \tag{2.2}$$

Thus, minimization of $\|\boldsymbol{\varepsilon}\|_R$ to give the Wilcoxon rank-based estimator of β can be obtained by calculating pairwise differences of the data as

$$\varepsilon_i - \varepsilon_j = (y_i - y_j) - \beta(x_i - x_j)\ .$$

Thus the rank-based estimator that minimizes

$$\widehat{\beta}_R = \text{Argmin } \textstyle\sum_{i<j}^{n} \left|\varepsilon_i - \varepsilon_j\right|$$

can be calculate by maximizing the likelihood

$$\mathrm{Y}_i - \mathrm{Y}_j \sim \text{Laplace}\left(\beta_1\left(x_i - x_j\right), \sigma\right),$$

where σ is a scale parameter. This will the likelihood specification in the Bayesian Wilcoxon rank-based estimation procedure.

The following theorem guarantees that using the Laplace distribution on pairwise differences as the likelihood in Bayesian leads to the Bayesian Wilcoxon rank-based estimate in linear model. This is a well-known result (Hettmansperger and McKean, 2011) but we will provide the proof here for completeness as this forms the basis of our approach.

Theorem 1.

$$\sum_{i<j} |\varepsilon_j - \varepsilon_i| \propto \sum_{i=1}^{n} \frac{R\left(\varepsilon_i\right)}{n+1}\varepsilon_i$$

Proof.

$$\sum_{i<j} |\varepsilon_j - \varepsilon_i| = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} |\varepsilon_j - \varepsilon_i| \tag{2.3}$$

$$= \frac{1}{2}\left[2(n-1)\,\varepsilon_{(n)} - 2\sum_{i=1}^{n-1}\varepsilon_{(i)} + 2(n-2)\varepsilon_{(n-1)} - 2\sum_{i=1}^{n-2}\varepsilon_{(i)} + \ldots + 2\varepsilon_{(2)} - 2\varepsilon_{(i)}\right] \tag{2.4}$$

$$= \frac{1}{2}\left[2(n-1)\,\varepsilon_{(n)} + 2(n-3)\varepsilon_{(n-1)} + \ldots + 2(-n+3)\varepsilon_{(2)} + 2(-n+1)\varepsilon_{(1)}\right] \tag{2.5}$$

$$= 2\sum_{i=1}^{n}\left(R\left(\varepsilon_{(i)}\right) - \frac{n+1}{2}\right)\varepsilon_{(i)} \tag{2.6}$$

$$= 2\sum_{i=1}^{n}\left(R\left(\varepsilon_i\right) - \frac{n+1}{2}\right)\varepsilon_i \propto \sum_{i=1}^{n}\frac{R\left(\varepsilon_i\right)}{n+1}\varepsilon_i \tag{2.7}$$

□

Dealing with the paired data will allow us to apply Bayesian Wilcoxon rank-based method directly on different models as long as we can write out the likelihood function. The limitation we have is that the models have to have additive errors and the responses have to be continuous. Over the last two decades, there has been an "MCMC revolution" in which Bayesian methods have become a highly popular and effective tool for the applied statistician. With the development of software like JAGS, logistic regression, mixed effect regression, Cox model or any other models can be easily arranged in Bayesian analysis.

2.2 Standard Error Correction for Dependence

Assuming the errors $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed according to a distribution with location 0 and finite variance σ^2, we can obtain the covariance structure of the paired responses $y_i - y_j$ for all $j > i$. We can show that the structure of the covariance matrix looks like Table 2.1.

	$y_1 - y_2$	$y_1 - y_3$	$y_1 - y_4$	$y_2 - y_3$	$y_2 - y_4$	$y_3 - y_4$
$y_1 - y_2$	$2\sigma^2$	σ^2	σ^2	$-\sigma^2$	$-\sigma^2$	0
$y_1 - y_3$	σ^2	$2\sigma^2$	σ^2	σ^2	0	$-\sigma^2$
$y_1 - y_4$	σ^2	σ^2	$2\sigma^2$	0	σ^2	σ^2
$y_2 - y_3$	$-\sigma^2$	σ^2	0	$2\sigma^2$	σ^2	$-\sigma^2$
$y_2 - y_4$	$-\sigma^2$	0	σ^2	σ^2	$2\sigma^2$	σ^2
$y_3 - y_4$	0	$-\sigma^2$	σ^2	$-\sigma^2$	σ^2	$2\sigma^2$

Table 2.1: An example of the covariance matrix when sample size is four

It can be considered as an unstructured covariance matrix, but there are some patterns we can use to form up the general case. It is clear that the diagonals are $2\sigma^2$, and some of them will be 0. The covariance between $y_i - y_j$ and $y_i - y_k$ is σ^2, while the covariance between $y_i - y_j$ and $y_j - y_k$ is $-\sigma^2$. By factoring out σ^2, the covariance matrix can be rewritten as the product of σ^2 and a structure matrix $\mathbf{A}$.

However, directly incorporating the covariance structure in Bayesian estimation of multivariate linear regression via a maximization of a Laplace likelihood is quite complex. First, we need to state the conditional likelihood and then find the appropriate conjugate prior to obtain the solution. It is similar to the univariate case of linear Bayesian regression. Consider the regression problem in matrix form, which is $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$, we can write our conditional likelihood as

$$\rho\left(\mathbf{E}|\mathbf{\Sigma}\right) \propto |\mathbf{\Sigma}|^{-n/2} \exp\left(-\tfrac{1}{2}\mathrm{tr}\left(\mathbf{E}^{\mathrm{T}}\mathbf{E}\mathbf{\Sigma}^{-1}\right)\right)$$

Rewrite the error $\mathbf{E}$ in terms of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{B}$, and we can get

$$\rho\left(\mathbf{Y}|\mathbf{X}, \mathbf{B}, \mathbf{\Sigma}\right) \propto |\mathbf{\Sigma}|^{-n/2} \exp\left(-\tfrac{1}{2}\mathrm{tr}\left((\mathbf{Y} - \mathbf{X}\mathbf{B})^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\mathbf{B})\mathbf{\Sigma}^{-1}\right)\right)$$

According to Sinay and Hsu (2014), we can develop a conditional form for the priors:

$$\rho\left(\boldsymbol{\beta},\boldsymbol{\Sigma}\right)=\rho\left(\boldsymbol{\Sigma}\right)\rho\left(\boldsymbol{\beta}|\boldsymbol{\Sigma}\right)$$

where $\rho\left(\boldsymbol{\Sigma}\right)\sim\mathcal{W}^{-1}\left(\mathbf{V}_0,\boldsymbol{\nu}_0\right)$, $\rho\left(\boldsymbol{\beta}|\boldsymbol{\Sigma}\right)\sim N\left(\boldsymbol{\beta}_0,\boldsymbol{\Sigma}\otimes\boldsymbol{\Lambda}_0^{-1}\right)$, $\otimes$ denotes the Kronecker product, and $\boldsymbol{\Lambda}_0$ is the prior precision matrix.

Then the posterior distribution can be expressed by using the above prior distribution and the likelihood:

$$\begin{aligned}\rho\left(\boldsymbol{\beta},\boldsymbol{\Sigma}_\epsilon|\mathbf{Y},\mathbf{X}\right) &\propto |\boldsymbol{\Sigma}_\epsilon|^{-(\boldsymbol{\nu}_0+2)/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{V}_0\boldsymbol{\Sigma}_\epsilon^{-1}\right)\right)\\ &\times |\boldsymbol{\Sigma}_\epsilon|^{-1}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\left(\mathbf{B}-\mathbf{B}_0\right)^{\mathrm{T}}\boldsymbol{\Lambda}_0\left(\mathbf{B}-\mathbf{B}_0\right)\boldsymbol{\Sigma}_\epsilon^{-1}\right)\right)\\ &\times |\boldsymbol{\Sigma}_\epsilon|^{-n/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left((\mathbf{Y}-\mathbf{X}\mathbf{B})^{\mathrm{T}}(\mathbf{Y}-\mathbf{X}\mathbf{B})\boldsymbol{\Sigma}_\epsilon^{-1}\right)\right)\end{aligned}$$

Sinay and Hsu (2014) proposed a more useful form of this posterior

$$\begin{gathered}\rho\left(\boldsymbol{\beta},\boldsymbol{\Sigma}|\mathbf{Y},\mathbf{X}\right)\propto|\boldsymbol{\Sigma}|^{-(\nu_0+n+2)/2}\\ \times\exp\left(-\tfrac{1}{2}\mathrm{tr}\left(\left(\mathbf{V}_0+\left(\mathbf{Y}-\mathbf{X}\mathbf{B}_\mathbf{n}\right)^{\mathrm{T}}\left(\mathbf{Y}-\mathbf{X}\mathbf{B}_\mathbf{n}\right)+\left(\mathbf{B}_\mathbf{n}-\mathbf{B}_0\right)^{\mathbf{T}}\boldsymbol{\Lambda}_0\left(\mathbf{B}_\mathbf{n}-\mathbf{B}_0\right)\right)\boldsymbol{\Sigma}^{-1}\right)\right)\\ \times|\boldsymbol{\Sigma}|^{-1}\exp\left(-\tfrac{1}{2}\mathrm{tr}\left(\left(\mathbf{B}-\mathbf{B}_{\mathrm{n}}\right)^{\mathrm{T}}\left(\mathbf{X}^T\mathbf{X}+\boldsymbol{\Lambda}_0\right)\left(\mathbf{B}-\mathbf{B}_{\mathrm{n}}\right)\boldsymbol{\Sigma}^{-1}\right)\right)\end{gathered}$$

Since it takes the form of a Matrix normal distribution times a inverse-Wishart distribution, we can get

$$\rho\left(\mathbf{B}|\mathbf{Y},\mathbf{X},\boldsymbol{\Sigma}\right)\sim\mathcal{MN}\left(\mathbf{B}_n,\boldsymbol{\Lambda}_n^{-1},\boldsymbol{\Sigma}\right)\ ,$$

where $\mathbf{B}_n=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\boldsymbol{\Lambda}_0\right)^{-1}\left(\mathbf{X}^{\mathrm{T}}\mathbf{Y}+\boldsymbol{\Lambda}_0\mathbf{B}_0\right)$ and $\boldsymbol{\Lambda}_n=\mathbf{X}^{\mathrm{T}}\mathbf{X}+\boldsymbol{\Lambda}_0$.

In our case, $\boldsymbol{\Sigma}$ is known if σ can be estimated. To the best of our knowledge, the idea that replacing the normal distribution with Laplace distribution in the above setup to get the posterior distribution for $\mathbf{B}$ is straightforward. Then, we can correct the biased estimator of $Cov(\hat{\beta})$, but this correction violates the idea that we want a method which is practically solvable in JAGS, so that it can be applied on different models directly. That is why we decide to use the sandwich variance to get the right $Cov(\widehat{\boldsymbol{\beta}})$.

With this covariance issue, $\widehat{\boldsymbol{\beta}}$ is still asymptotically unbiased but it is no longer efficient. Huber (1967) and White (1980) first introduced the sandwich estimator. The naive estimate of $Cov(\hat{\beta})$ is $\left(X^{\mathrm{T}}\mathbf{\Sigma}^{-1}X\right)^{-1}$, where $\mathbf{\Sigma}$ is the covariance matrix. Essentially, Bayesian Wilcoxon rank-based estimate is equivalent to the LAD solution based on the paired data. Thus we replace Σ by the product of τ_S^2 and the structure matrix $\mathbf{A}$, where τ_S^2 is a scale parameter that can be estimated from the data (Koul et al., 1987). Thus $Cov(\hat{\beta}) \approx \tau_S^{-2}\left(X^{\mathrm{T}}\mathbf{A}^{-1}X\right)^{-1}$. More details on τ_S are given in Section 4.3.

2.3 Real Data Examples

In order to demonstrate the application in linear regression model, we offer two examples to compare the Bayesian Wilcoxon estimates and the rank-based estimates of the regression coefficients by using Wilcoxon scores.

We start with a simple set of regression data and proceed to multiple regression issues. The response to this data set is the number of international telephone calls made in Belgium from 1950 to 1973 (ten of millions). From year 1964 to 1969, the data was heavily contaminated because of using a different recording system. The system provided the total number of minutes of these calls and this is why the values of Y from year 1964 to 1969 are much larger than it should be. Time, the years, is our only variable predictor. The data are discussed in Rousseeuw and Leroy (1987).

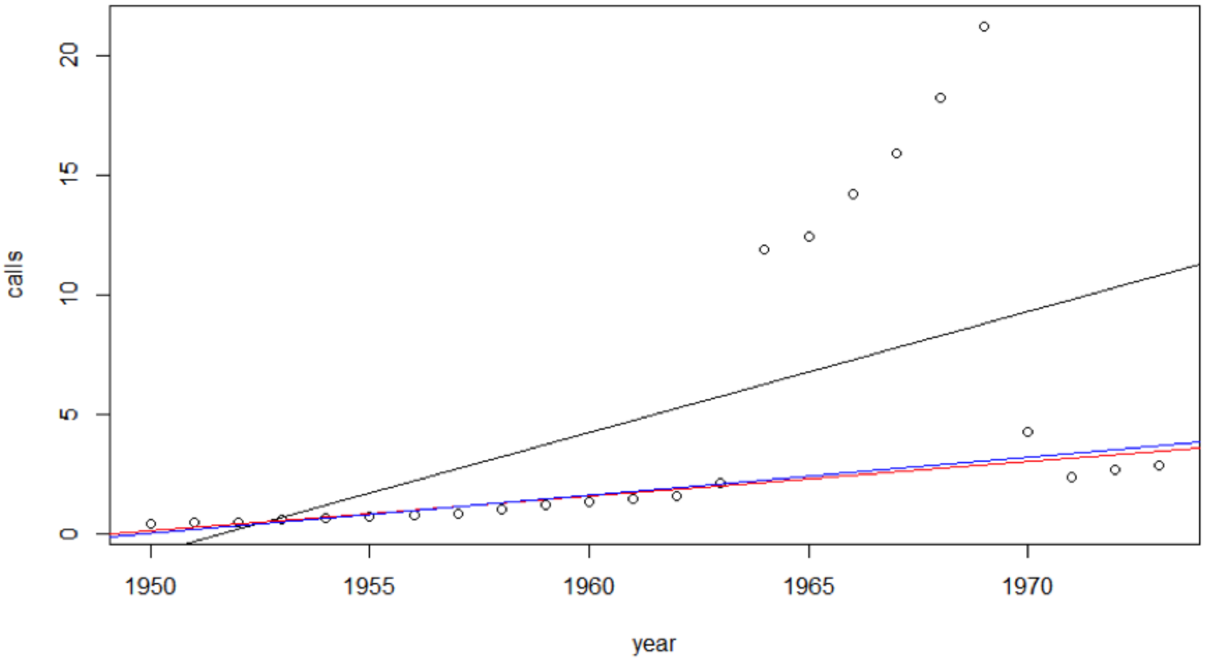

Figure 2.1: LS, Wilcoxon and Bayesian Wilcoxon fits in telephones calls data

The traditional Wilcoxon rank-based estimates of the intercept and slope can be found using the R package `Rfit`. These are -7.13 and 0.145, respectively, while the LS estimates are -26 and 0.504. In Figure 2.1 we give the scatter-plot of the data overlaid with the LS, Wilcoxon and Bayesian Wilcoxon fits. It is clear that the years 1964 through 1969 had a profound effect on the LS fit (black line) while the Wilcoxon fit (blue line) and Bayesian Wilcoxon fit (red line) were much less sensitive to these years, and the Bayesian Wilcoxon estimate is still unbiased even if we ignore the covariance causing by pairing the original data. Regarding to the uncertainty, $Cov(\hat{\beta})$ is changed from 0.0304 to 0.0273 after using the correction of sandwich variance. We can see that the years 1963 and 1970 were also partially affected by the new recording system, which means about a quarter to one third of the data are outliers. The robustness of Bayesian Wilcoxon estimate is well demonstrated by this example.

As a large data set, we consider the data on the salaries of professional baseball pitchers for the 1987 baseball season. This data set was taken from the data set on baseball salaries, which was used in the 1988 ASA Graphics Section Poster Session. It can be obtained at the web site `http://lib.stat.cmu.edu/datasets`. The dataset has the salary data of 176 pitchers at the beginning of 1987 as the response variable. We consider the career summary statistics from 1986 professional baseball season as the predictors. They are years in professional baseball, average of wins per year, average of losses per year, earned run average, average of games per year, average innings per year and average saves per year. It is a well known dataset which contains several large outliers. There are three identified outliers corresponding to the pitchers Rick Sutcliff, Phil Niekro and Steve Carlton. They are three outstanding pitchers, but their salaries are low because they are at the end of careers (more than 20 years of pitching). However, the number of years should be positively correlated to the salary.

As we can see in Table 2.2, Bayesian Wilcoxon rank-based fit provided similar results with much smaller standard deviations for all of the coefficients. The corrected standard deviations (CSD) are showed in the last column. Taking the effect caused by the covariance into consideration may increase the uncertainty of the estimate, but the

	Estimate (Rank)	SD (Rank)	Estimate (BRank)	SD (BRank)	CSD (BRank)
intercept	4.2192	0.3246	4.418	0.2677	0.1995
logYears	0.8391	0.0438	0.8282	0.0054	0.0254
aveWins	0.0451	0.0277	0.0387	0.0030	0.0101
aveLosses	-0.0242	0.0263	-0.0305	0.0031	0.0631
era	-0.1461	0.0691	-0.1599	0.0078	0.0631
aveGames	0.0061	0.0039	0.0052	0.0004	0.0002
aveInning	0.0042	0.0026	0.0037	0.0003	0.0001
aveSaves	0.0121	0.0113	0.0102	0.0012	0.0017

Table 2.2: Comparison results based on telephone calls data

uncertainty is still smaller than normal rank estimate. In particular, the predictor, years in professional baseball is the most important factor among others. This is the main reason why the three players we mentioned before with large values of years but relevant low salaries should be considered as influential outliers.

Table 2.3 shows the results of a cross-validation study. An eight-fold cross-validation was performed to compare the predictive performance of the Bayesian Wilcoxon rank-based estimator (BRank) and the normal rank-based estimator. The average of the prediction errors for BRank estimate is slightly smaller, but among those eight folds, BRank estimate provides smaller prediction errors in six folds. In this example, BRank estimate shows its robustness and efficiency comparing with normal rank-based estimate.

	F1	F2	F3	F4	F5	F6	F7	F8	Average
BRank	0.839	0.110	0.367	0.247	0.222	0.147	0.282	0.103	0.290
Rank	0.863	0.104	0.336	0.267	0.226	0.150	0.294	0.110	0.294

Table 2.3: Comparison results based on baseball pitcher data

Chapter 3

Linear Model with Measurement Error

3.1 Introduction to the measurement error modeling

In many research situations, measurement error can occur and occurs when a variable of interest cannot be accurately observed, but is observed with error instead. Measurement error, also referred as errors-in-variables (Stefanski, 2000), may occur in a variety of circumstances, including, but not limited to miscalibration of a measuring instrument, sampling error, misclassification, modeling of abundance estimates, and whenever the analyzed responses are estimated. The effects of ignoring measurement error in analyzes were well documented, including bias in parameter estimation and power loss. (Carroll *et al.*, 2006). Measurement error may be classified as either Berkson or classical in a continuous random variable. Berkson measurement error (Berkson, 1950) occurs when a researcher attempts to achieve a target (V_i) value, but attain a true (X_i) value instead. This work will focus on classical measurement error that occurs when the true value of the continuous random variable (X_i) is measured with error resulting in an observed value (V_i). The classic additive error measurement model for the wrong random variable $V_i = X_i + U_i$, where U_i represents the measurement error, $E\left(U_i|x_i\right) = 0$, and $\mathrm{Var}\left(U_i|x_i\right) = \sigma_u^2$. As result of mean 0, additive error structure means that V is an unbiased estimator of X since $E\left(V_i|x_i\right) = x_i$. Although the focus of this chapter is on the additive error structure, there are other mechanisms for measurement errors to occur. For example, the measurement error structure can be linear $V_i = \theta_0 + \theta_1 X_i + U_i$ or multiplicative $V_i = X_i * U_i$.

In order to capture measurement errors when present, the measurement error variance σ_u^2 is known or estimated from replicate data (repeated independent observations on the same subject at the same time) must be assumed. There exist many ways to correct measurement errors, however a single method is unlikely to provide the best approach for every situation.

Moment-based error correcting method is one of the most straightforward methods that takes a linear transformation of the original estimates of the coefficients in the models. Buonaccorsi (2010) discussed many moment-based measurement error correction formulas for different modeling situations, including SLR (Simple Linear Regression). The moment-based formula for the measurement error corrected estimator of the slope in the SLR setting is

$$\hat{\beta}_{1,MOM} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_V^2 - \hat{\sigma}_u^2} = \frac{\hat{\sigma}_{XY}}{\hat{\sigma}_X^2},$$

where $\hat{\sigma}_{XY}$ is the estimated covariance between the observed response and unobserved explanatory variable, $\hat{\sigma}_v^2$ is the estimated variance of the mis-measured explanatory variable (V), and $\hat{\sigma}_u^2$ is the estimated measurement error variance, with $\hat{\sigma}_X^2 = \hat{\sigma}_v^2 - \hat{\sigma}_u^2$.

Cook and Stefanski (1994) introduced the SIMEX (Simulation-Extrapolation Estimation) measurement error accounting approach when the measurement error variance is known or reasonably well estimated. This method is a general framework applicable to many situations and can be found in the R package as '*simex*' (Lederer and Kchenhoff, 2019). The first step of the SIMEX procedure is to simulate pseudo data with an additional measurement error than what was actually observed. For instance, we can simulate additional data with measurement error variance $(1 + \theta)\sigma_u^2$ and study the relationship between regression estimates and θ. The SIMEX corrected estimate is the one at $\theta = -1$.

Bartlett and Keogh (2016) compare a Bayesian regression calibration approach and conclude that the Bayesian method has several statistical advantages over regression calibration techniques. Although a number of Bayesian methods exist to correct for measurement error in the explanatory variable, they all have the same underlying steps. Chapter 9 of Carroll *et al.* (2006) focuses on Bayesian error measurement models and describes five general steps in the development of a Bayesian error measurement model:

(i) specify the likelihood model as if X were observed, (ii) select the proper measurement error model (additive, linear, multiplicative, etc.), (iii) form the likelihood function as if X were observed, (iv) select compatible prior distributions, and (v) compute the complete conditional distributions.

In a Bayesian analysis where measurement error appears in the explanatory variable, the true unobserved value (X_i) is treated like a latent variable and given a prior distribution. Carroll *et al.* (2006) describes the Bayesian SLR measurement error model using three models: an outcome model, a measurement model, and an prior model.

The outcome model is a model for the outcomes that you would obtain if measurement error was not present and is

$$Y_i \sim N\left(\beta_0 + \beta_1 X_i, \sigma_\epsilon^2\right) \tag{3.1}$$

The measurement model can be described for the mis-measured variable given the true variable and is

$$V_i\,|X_i \sim N\left(X_i, \sigma_u^2\right)$$

This is a direct result of Equation 3.1, which shows that when classical additive measurement error is part of the explanatory variable, $E\left(V_i|x_i\right) = x_i$ and $\text{Var}\left(V_i|x_i\right) = \sigma_u^2$.

The prior model is the model when the true unobserved variable (x_i) and, by centering the observed variable is

$$X_i \sim N\left(0, \sigma_x^2\right)$$

In this model, $\sigma_{\mathbf{u}}^2$ is assumed known and all other parameters $(\beta_0, \beta_1, \tau_X = \frac{1}{\sigma_x^2}, \tau_\epsilon = \frac{1}{\sigma_\epsilon^2})$ are given prior distributions, which depend on the particular situation and are often selected as non-informative.

Jureckova et al. (2016) evaluated the effects of measurement errors on rank-based estimators of linear model parameters. They showed that the local asymptotic bias of rank-based estimators does not depend on the rank test score-generating functions selected or on the unknown distribution of the model errors. It only depends on the slope parameter vector value and the covariance matrix of the regressor error distribution.

This result gives us the opportunity to introduce desirable properties of non-parametric methods for measurement error problems. Our goal here is to provide a much simpler approach for rank-based estimator in the presence of measurement error instead of looking for the asymptotic distribution for the coefficients. Using Bayesian methodology gives us a convenient way of getting rank-based inference for different models without figuring out the specific asymptotic distributions.

3.2 Bayesian Wilcoxon Rank-based Method in Linear Model with Measurement Error

Fortunately, our nonparametric Bayesian method starts from a likelihood function as well. Even with measurement error presenting, the first step is still specifying a likelihood model, which is $Y^* \sim \text{Laplace}\,(\beta_0 + \beta_1 X^*, \Sigma^*)$, where Y^* and X^* represent the pairwise differenced data. Since we are only focusing on the classical model of measurement error, given the observable data V, we need to specify a distribution for the unobserved X. In general Bayesian approach, this will be the same as the measurement error model under measurement errors. The measurement error U_i is often assumed to follow the normal distribution with mean zero and a known variance σ_u^2

$$E\,(U_i|x_i) = 0 \text{ and } \text{Var}\,(U_i|x_i) = \sigma_u^2.$$

Under this setting, the measurement error model is:

$$V_i|X_i \sim N\,(X_i, \sigma_u^2).$$

The typical Bayesian approach treats X as missing data, and in effect, by drawing from X's conditional distribution given all other variables, it imputes multiple times. Thus, as if X were available, the likelihood of all the data, including V, is formed. On the other hand, parameters are treated with probability methods as if they were random, one of the essential differences. If we are to treat parameters as random, then prior distributions need to be given. Much of the controversy regarding Bayesian methods among statisticians revolves around these previous distributions. For example, in this work, X's variance has a gamma distribution as a prior.

3.3 Posterior Inference

Bayesian inference is based on the posterior density which, given the observed data, is the conditional density of unobserved quantities (parameters and unobserved data), and summarizes all the information about the unobserved quantities. To calculate the posterior, we can take the data and parameters' joint density and integrate out the parameters, at least in principle, to obtain the data's marginal density. In order to obtain the posterior density, we can then divide the joint density by this marginal density. There are many "textbook examples" where the marginal can be calculated analytically, but this is often a non-trivial problem requiring high-dimensional numerical integration in practical applications. Much recent research has focused on the computational problem. The Gibbs sampler is the method that currently receives the most attention in the literature.

The Gibbs sampler, often referred to as Markov Chain Monte Carlo (MCMC), generates a Markov chain, the posterior distribution of which is stationary. The key feature of the Gibbs sampler is that this chain can be simulated using only the joint density of the parameters, the non-observed X-values and the observed data, such as the product of likelihood and prior, and not the unknown posterior density that requires an integral that is often intractable. If the chain runs long enough, the observations in a chain sample are distributed approximately the same, with a common distribution equal to the posterior. Thus it is possible to estimate posterior moments, posterior density and other posterior quantities from a chain sample. Due to the observed data and other parameters, the Gibbs sampler "fills in" or imputes the values of the unobserved X covariates by sampling from their conditional distribution. This kind of imputation differs in two important ways from the imputation of regression calibration. First, a large number of imputations are made by the Gibbs sampler from the conditional distribution of X, whereas the regression calibration uses a single imputation, namely the conditional expectation of X given v. Second, when imputing X values, the Gibbs sampler conditions on both Y and V, but when imputing X, regression calibration does not use Y information.

For a simple linear model with the additive classical measurement error, the expression of three sub-models are as follow

$$f\left(\mathbf{Y}|\mathbf{X},\boldsymbol{\beta}\right)\text{: response model with parameters } \boldsymbol{\beta}=(\beta_0,\beta_1,\sigma_\epsilon^2)$$
$$f\left(\mathbf{V}|\mathbf{X},\boldsymbol{\lambda}\right)\text{: measurement error model with parameters } \boldsymbol{\lambda}=(\sigma_{\mathbf{u}}^2)$$
$$f\left(\mathbf{X}|\boldsymbol{\pi}\right)\text{: prior model with parameters } \boldsymbol{\pi}=(\mu_x,\sigma_x^2)$$

An important assumption is that the measurement error is non-differential; that is, the size of the measurement error is independent of the response. Then the joint distribution will be

$$\begin{gathered} f\left(\mathbf{Y},\mathbf{X},\mathbf{V},\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{\pi}\right) \\ = f\left(\boldsymbol{\beta}\right) f\left(\boldsymbol{\lambda}\right) f\left(\boldsymbol{\pi}\right) \prod_{i=1}^{n} f\left(Y_i|X_i,\boldsymbol{\beta}\right) \prod_{i=1}^{n} f\left(V_i|X_i,\boldsymbol{\lambda}\right) \prod_{i=1}^{n} f\left(X_i|\boldsymbol{\pi}\right) \end{gathered}$$

Firstly, our job is to derive the full conditional posterior distributions for all the unknown values in order to utilize the Gibbs sampler. Where we normally have unknown parameters apriori independent. Given the initial values $\theta^{(0)}$ and $\mathbf{X}^{(0)}$ based on the Gibbs sampler, we first draw the samples of the unobserved X from its full conditional posterior distribution (3.2). The full conditional posterior distributions of unknown data X can be written as

$$\begin{aligned} f\left(X_i|Y_i,V_i,\boldsymbol{\theta}\right) &\propto f\left(Y_i|X_i,\beta\right) f\left(V_i|X_i,\lambda\right) f\left(X_i|\pi\right) \\ &= f\left(Y_i|X_i,\beta_0,\beta_1,\sigma_\epsilon^2\right) f\left(V_i|X_i,\sigma_u^2\right) f\left(X_i|\mu_x,\sigma_x^2\right) \qquad (3.2) \end{aligned}$$

where $\boldsymbol{\theta}=\beta,\lambda,\pi$.

Then the samples of the unknown parameters $\boldsymbol{\theta}$ can be generated from (3.3) and (3.4). The full conditional posterior distributions of unknown parameters $\boldsymbol{\theta}$ is

$$\begin{aligned} f\left(\beta_j|\beta_{\setminus j}),\sigma_\epsilon^2,\mathbf{Y},\mathbf{X}\right) &\propto f\left(\mathbf{Y}|\mathbf{X},\boldsymbol{\beta}\right) f\left(\beta_j|\mathbf{X},\sigma_\epsilon^2\right) \\ &= \prod_{i=1}^{n} f\left(\beta_j\right) f\left(Y_i|X_i,\beta_0,\beta_1,\sigma_\epsilon^2\right) \qquad (3.3) \\ f\left(\sigma_\epsilon^2|\mathbf{X},\mathbf{Y},\boldsymbol{\theta}\right) &\propto f\left(\mathbf{Y}|\mathbf{X},\boldsymbol{\beta}\right) f\left(\sigma_\epsilon^2|\boldsymbol{\beta},\mathbf{X}\right) \end{aligned}$$

$$= \prod_{i=1}^{n} f\left(\sigma_\epsilon^2\right) f\left(Y_i|X_i, \beta_0, \beta_1, \sigma_\epsilon^2\right) \tag{3.4}$$

After repeating the above procedures repeatedly for a burn-in period, we are finally able to obtain the desired samples from the posterior distribution (3.5). The joint posterior densities of the unknown values can be written as

$$f(\mathbf{X}, \boldsymbol{\theta}|\mathbf{Y}, \mathbf{V}) \propto f\left(\boldsymbol{\beta}\right) f\left(\boldsymbol{\lambda}\right) f\left(\boldsymbol{\pi}\right) \prod_{i=1}^{n} f\left(X_i|\boldsymbol{\pi}\right) \prod_{i=1}^{n} f\left(Y_i|X_i, \boldsymbol{\beta}\right) \prod_{i=1}^{n} f\left(V_i|X_i, \boldsymbol{\lambda}\right) \tag{3.5}$$

We will now derive the Bayesian formulation for the Wilcoxon rank-based estimation in measurement error linear regression. For obtaining the Bayesian Wilcoxon rank-based estimation, the joint distribution will depend on distributions of pairwise differences of random variables and can be written as

$$f(\mathbf{Y}, \mathbf{X}, \mathbf{V}, \beta, \lambda, \pi)$$

$$= f(\beta) f(\lambda) f(\pi) F_{i<j}\left((Y_i - Y_j)|(X_i - X_j), \beta\right) \times$$

$$F_{i<j}\left((V_i - V_j)|(X_i - X_j), \lambda\right) F_{i<j}\left((X_i - X_j)|\pi\right)$$

The full conditional posterior distributions of unknown data X will be

$$f\left((X_i - X_j)|(Y_i - Y_j), (V_i - V_j), \theta\right)$$

$$\propto f\left((Y_i - Y_j)|(X_i - X_j), \beta\right) f\left((V_i - V_j)|(X_i - X_j), \lambda\right) f\left((X_i - X_j)|\pi\right)$$

$$= f\left((Y_i - Y_j)|(X_i - X_j), \beta_0, \beta_1, \sigma_\epsilon^2\right) f\left((V_i - V_j)|(X_i - X_j), \sigma_u^2\right) f\left((X_i - X_j)|\mu_x, \sigma_x^2\right)$$

The full conditional posterior distributions of unknown parameters now becomes

$$f\left(\beta_k|\beta_{\backslash k}, \sigma_\epsilon^2, \mathbf{Y}^*, \mathbf{X}^*\right)$$

$$\propto f\left(\mathbf{Y}^*|\mathbf{X}^*, \boldsymbol{\beta}\right) f\left(\beta_k|\mathbf{X}^*, \sigma_\epsilon^2\right)$$

$$= \prod_{i<j} f\left(\beta_k\right) F_{i<j}\left((Y_i - Y_j)|(X_i - X_j), \beta_0, \beta_1, \sigma_\epsilon^2\right)$$

$$f\left(\sigma_\epsilon^2|\mathbf{X}^*, \mathbf{Y}^*, \boldsymbol{\theta}\right)$$

$$\propto f\left(\mathbf{Y}^*|\mathbf{X}^*, \boldsymbol{\beta}\right) f\left(\sigma_\epsilon^2|\beta, \mathbf{X}^*\right)$$

$$= \prod_{i<j} f\left(\sigma_\epsilon^2\right) F_{i<j}\left((Y_i - Y_j)|(X_i - X_j), \beta_0, \beta_1, \sigma_\epsilon^2\right)$$

Finally, the joint posterior densities of the unknown values can be written as

$$f(\mathbf{X}^*, \boldsymbol{\theta}|\mathbf{Y}^*, \mathbf{V}^*)$$

$$\propto f\left(\boldsymbol{\beta}\right) f\left(\boldsymbol{\lambda}\right) f\left(\boldsymbol{\pi}\right) F_{i<j}\left((X_i - X_j)|\boldsymbol{\pi}\right) \times$$

$$F_{i<j}\left((Y_i - Y_j)|(X_i - X_j), \boldsymbol{\beta}\right) F_{i<j}\left((V_i - V_j)|(X_i - X_j), \boldsymbol{\lambda}\right)$$

where X^*, Y^* and V^* are calculated by taking paired differences of the original X, Y and V, respectively. Here F denotes the joint distribution.

To obtain a joint distribution $F(a_1, \ldots, a_p)$ with full conditionals $f_1, \ldots, f_p$ where f_j is the distribution of a_j conditional on $(a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_p)$, the Gibbs sampler simulates successively from all conditionals, changing one component of $\boldsymbol{a}$ at a time. The Gibbs sampler works as follows.

- We begin some initial value $\boldsymbol{a}^{(0)} = \left(a_1^{(0)}, \ldots, a_p^{(0)}\right)$.
- Then we start iteration t: Given $\left(a_1^{(t-1)}, \ldots, a_p^{(t-1)}\right)$, update

  1. $a_1^{(t)}$ according to $\pi_1\left(a_1|a_2^{(t-1)}, \ldots, a_p^{(t-1)}\right)$,
  2. $a_2^{(t)}$ according to $\pi_2\left(a_2|a_1^{(t)}, a_3^{(t-1)}, \ldots, a_p^{(t-1)}\right)$,

  ...

  p. $a_p^{(t)}$ according to $\pi_p\left(a_p|a_1^{(t)}, \ldots, a_{p-1}^{(t)}\right)$

We implemented this in R and JAGS through the package `rjags`. In the next section we will use this for conducting Monte Carlo simulation experiments under various measurement error as well as model error distribution scenarios.

3.4 Simulation

For the Bayesian method, JAGS (Plummer, 2003) was used to implement the Gibbs sampling algorithm (Geman and Geman, 1984), resulting in samples from the joint posterior distribution of the parameters. We ran three independent chains, each with random starting values. In our study, the number of samples for adaption phase, the number of samples to discard as burn-in and the number of MCMC steps retained differ from case to case. The number of iterations and thinning rate were determined by the trace plots in order to guarantee convergence.

3.4.1 Simulation Scenario 1: Heavy Tails

In the first simulation scenario, we considered the simplest linear model $Y = \beta X$, where the true data are generated as $X\sim$ Unif (-1.5, 1.5) and the true parameter is taken as $\beta = 1$. We will compare our method with classical Bayesian approach, moment-based error correcting method, and the SIMEX method. Our method required longer adaption phase and larger size of burn-in in comparison to the classical Bayesian approach. So, we considered these values to make ensure both approaches gave favorable convergence results. In order to see the effects caused by different levels of measurement error, we three different variances for the measurement error distribution ($\sigma_u^2 = 0.1, 0.3, 0.5$) representing increasing levels of measurement error. As we mentioned in Section 2.2, Bayesian Wilcoxon rank-based estimate is the LAD solution based on paired data. Thus, the variance of the new measurement error for the paired x should be two times the original measurement variance. To see the robustness brought by Bayesian Wilcoxon rank-based, five different t distribution model errors were considered, with degrees of freedom from 1 to 5 representing decreasing tail-thickness of the model error distribution. These represent model error distributions from heavy-tailed (Cauchy) to moderate taled. We considered a sample of size $n = 30$ and we conducted 50 repetitions for each case. Table 3.1 gives the estimated values and estimated variances.

σ_u^2	DF	MCM Mean (Var)	SIMEX Mean (Var)	Bayesian Mean (Var)	BRank Mean (Var)
0.1	1	-1.530(200.970)	-1.270(152.370)	0.000(23.540)	1.003(0.191)
	2	0.912(0.251)	0.913(0.254)	0.920(0.247)	0.908(0.096)
	3	0.991(0.099)	0.991(0.100)	0.973(0.085)	1.020(0.070)
	4	0.964(0.092)	0.962(0.092)	0.963(0.092)	0.963(0.087)
	5	0.946(0.070)	0.947(0.070)	0.960(0.070)	0.933(0.077)
0.3	1	-2.110(324.120)	-1.530(188.550)	-0.200(31.800)	0.941(0.209)
	2	0.898(0.273)	0.887(0.282)	0.903(0.265)	0.890(0.011)
	3	0.992(0.122)	0.980(0.126)	0.959(0.105)	1.046(0.105)
	4	0.965(0.103)	0.951(0.106)	0.953(0.100)	0.991(0.103)
	5	0.958(0.074)	0.957(0.077)	0.970(0.076)	0.971(0.088)
0.5	1	-2.600(460.130)	-1.890(256.130)	-0.520(63.990)	0.863(0.240)
	2	0.893(0.327)	0.830(0.318)	0.886(0.316)	0.883(0.156)
	3	0.998(0.177)	0.927(0.161)	0.941(0.146)	1.075(0.118)
	4	0.971(0.125)	0.904(0.121)	0.932(0.125)	1.039(0.147)
	5	0.978(0.093)	0.946(0.101)	0.983(0.101)	1.055(0.145)

Table 3.1: Mean and variance for β in Simulation 1

From Table 3.1, we observed that Bayesian Wilcoxon rank-based estimate (BRank) provided reasonable results under heavy-tailed model error distributions as expected. Compared to moment-based error correcting method and the SIMEX method, our proposed Bayesian Wilcoxon rank-based estimate was closer to the true β with smaller uncertainty in many cases. Even when the model error distributions are closer to the normal distribution ($df = 5$), the Bayesian Wilcoxon rank-based estimate did not lose much efficiency relative to the other methods. The performance of the SIMEX method was much worse when the variance of the measurement error increased form 0.1 to 0.5, while the Bayesian Wilcoxon rank-based estimate was not significantly affected by this change in measurement error. Moment-based correcting method and normal Bayesian approach were too sensitive to heavy-tailed model error distributions. When the model error followed the t-distribution with degree of freedom two, the mean of the estimated β are close to the true β, but the variance is very large unlike Bayesian Wilcoxon rank-based estimate. The results of the moment-based method, SIMEX, and the classical Bayesian approach did not give reasonable results for the t-distribution with 1 degree of freedom.

This is expected since the mean does not exist for this distribution. However, this highlights that our non-parametric approach can be applied in situations where the classical approaches cannot.

3.4.2 Simulation Scenario 2: Outliers

In the second simulation scenario, we want to demonstrate the robustness of our method in the presence of gross outliers. For this we created a mixed normal distribution for the model error. We again considered the simple linear model $Y = \beta X$, where the true data are generated as $X\sim$ Unif (-1.5, 1.5) and the true parameter is taken as $\beta = 1$. Once again, we will compare our method with classical Bayesian approach, moment-based error correcting method, and SIMEX. The error distribution was created as a Huber mixture of two normal distributions where 95% of the model errors were generated from $N(0, 1)$ and 5% of them were generated from $N(0, 10)$. We take a common measurement error distribution $N(0, 0.5)$ in this study. The sample sizes considered are $n = 30$ and $n = 50$ with 50 replications. The results of the simulation experiment are given in Table 3.2.

	MCM Mean (Var)	Simex Mean (Var)	Bayesian Mean (Var)	BRank Mean (Var)
30	0.961 (0.428)	0.887 (0.365)	0.961 (0.393)	1.023 (0.089)
50	0.976 (0.229)	0.922 (0.209)	0.948 (0.218)	0.944 (0.039)

Table 3.2: Mean and variance for β in Simulation 2

Similarly in Table 3.2, for moment-based error correcting method, the SIMEX method, and normal Bayesian approach, the means of the estimated β are close to the true β, but the variances are very large unlike the Bayesian Wilcoxon rank-based estimate. The estimated value of the Bayesian Wilcoxon rank-based estimate is closer to the true value of 1 for the smaller sample size ($n = 30$). It performs better than SIMEX but worse than the moment-based method for the larger sample size ($n = 50$). This simulation shows the robustness of Wilcoxon rank-based estimate in the presence of outliers, especially when the sample sizes are small. Moreover, the efficiency is far superior than the competition. In applications with real data, the percentage of outliers may be much much larger than

5%, so we need to study the performances of these methods with different percentage of outliers.

3.4.3 Simulation Scenario 3: Levels of Contamination

Our third simulation demonstrates the effect of different percentage of outliers on the performance of measurement error estimators. We again considered the simple linear model $Y = \beta X$, where the true data are generated as $X\sim$ Unif (-1.5, 1.5) and the true parameter is taken as $\beta = 1$. Again, we will compare our method with classical Bayesian approach, moment-based error correcting method, and SIMEX. The error distribution was created as a Huber mixture of two normal distributions as discussed in Section 3.4.2. However, we considered four cases in total with 1%, 5%, 10% and 15% outliers generated from $N(0, 10)$. The sample size is set to be 50, with 50 runs of simulations. The results for these four cases are given in Table 3.3.

Contamination	MCM Mean (Var)	SIMEX Mean (Var)	Bayesian Mean (Var)	Bayesian-R Mean (Var)
1%	0.936 (0.094)	0.879 (0.080)	0.909 (0.085)	0.979 (0.036)
5%	0.976 (0.229)	0.922 (0.209)	0.948 (0.218)	0.944 (0.039)
10%	1.052 (0.538)	0.984 (0.459)	1.022 (0.515)	0.916 (0.035)
15%	1.211 (0.583)	1.149 (0.545)	1.191 (0.589)	0.959 (0.058)

Table 3.3: Mean and variance for β in Simulation 2

Under the assumption that outlier was not a big issue, for example, with the presence of 1% outliers, the performance of the Bayesian Wilcoxon rank-based estimate is surprisingly the best among all these four methods. When the percentage of the outliers increased to 10%, the Bayesian Wilcoxon rank-based estimate is not the best estimate due to the bias on estimated β, but the variance did not increase as compared to the 1% outliers case. As expected, Bayesian Wilcoxon rank-based was the best choice when the data contain 15% outliers, and the biases for the rest three estimations were much larger.

3.4.4 Summary of Simulation Results

The moment-based error correcting method, SIMEX, and the classical Bayesian approach had noticeable performance issues when the variance of the measurement error increased. SIMEX method only provided an estimated β with smallest mean of bias in the 10% outliers scenario, but the variance was increased substantially compared to the 1% outliers case. The moment-based error correcting method is only competitive in the 1% outliers case and the 5% outliers case, while classical Bayesian approach is only competitive in the 5% outliers case. Otherwise, in all the cases where there is a heavier tail, a larger amount of contamination, or a smaller sample size, the Bayesian Wilcoxon rank-based estimator gave superior performance than all the existing methods.

To conclude, our method showed its advantage when dealing with heavy tailed model error distributions and data that containing outliers with extreme large values. If the outliers play an important role in the data, the Bayesian Wilcoxon rank-based estimate provided robust and efficient estimates. Compared to the regular R-estimate proposed by Jureckova et al. (2016), our method is much more computationally efficient and flexible. This is achieved by utilizing a Bayesian estimation framework. Problems caused by measurement error were resolved in the Bayesian iterations not after a preliminary fit and the scale parameter for rank-based estimate is also estimated directly in the MCMC iteration. This is an added advantage of the Bayesian formulation and more details about the scale parameter will be discussed in Section 4.3.

Chapter 4

Single Index Model

4.1 Introduction to the Single Index Model

Utilizing a potential lower-dimensional structure of a linear regression function holds the way to important derivation for many studies where the assumptions of traditional linear regression are failed. Single index model (SIM) is an exceptionally well known semi-parametric model to give a basic and interpretable system for understanding an intricate connection between a response variable Y_i and its $(p \times 1)$ dimensional covariate vector X_i, for $p > 1$. Single-index models offer adaptable option in contrast to standard linear regression, with the conditional expectation for the reaction of Y_i allowed to be an discretionary link function of a finite linear combination of predictors: $E\left(Y_i|X_i\right) = g\left(X_i'\boldsymbol{\beta}\right)$, $i = 1, ..., n$. The vector of regression coefficients β, also called index vector, is recognizable up to a steady of proportionality. The link function g is viewed as an infinite-dimensional irritation parameter. Such models emerge in Friedman and Stuetzle's (1981) projection pursuit regression, and they have broad applications in econometrics.

A SIM obviously provides a pragmatic compromise between a fully nonparametric regression and a completely parametric multiple linear regression. It significantly renovate from a linear model to oblige both non-linear main effects and higher order interactions of the covariate effects. In addition, different from the completely nonparametric multiple regression function, SIM provides a distinct understanding of the overall significance of the predictor effects via the magnitudes of the index coefficients, $\beta = (\beta_1, \beta_2, \cdots, \beta_p)$. This is exceptionally needful in biomedical studies to comprehend the complex effectors of predictors and to try and assess the locally linear effects.

During the past decades, there were numbers of remarkable literatures for SIM method. The rapid development during the last decade. Antoniadis et al. (2004) wrote an prominent reviews of these methods. There are two categories of existing methods. The first one is Average derivative method, e.g., Powell et al. (1989); the method uses weighted gradient $(\partial/\partial x)g(x) = \beta g'(\beta' x)$ to estimate β. In any case, they require exceptionally restrictions conditions to accomplish the consistency of the estimator of α. This method uses kernel smoothing to estimate $g(\cdot)$, which will lead to the curse of dimensionality even when the number of predictors p is only moderately high. The other category of methods is M-estimation, e.g., Hardle et al. (1993); M-estimation based approaches have good asymptotic properties. Nevertheless, regardless of good hypothetical properties, the semiparametric approach regularly prompts computational difficulties when attempting to evaluate the estimate of the index vector, to which requires the solution to a high-dimensional optimization problems. Producing point estimates of β and the link function $g(\cdot)$ with great empirical and asymptotic properties has been discussed a lot in existing literature on SIM. However, in most cases, it is hard to find out a realistic evaluation of the uncertainty under the predictions and the estimates in real applications. Even in moderate number of dimension cases, frequentist approaches may fail to capture the uncertainty.

Bayesian inference has been successful with many nonlinear regression models. For the application In SIM, it was first mentioned by Antoniadis *et al.* (2004). The link function $g(\cdot)$ is usually treated as a non-parametric function, however, the frequentist approaches are not. A basis representation, wavelets and Gaussian process prior on $g(\cdot)$ are the three popular methods to model the link function. For example, Antoniadis *et al.* (2004) uses B-splines which is a basis representation, Park *et al.* (2005) proposed a wavelets related method and Choi *et al.* (2011) modeled $g(\cdot)$ by using a Gaussian process prior. All these three ways have their own problems. Selecting the best number of the knots poses computational issues while using splines, because cross-validation is the only way we can ensure the optimal solution. Similarly, choosing the number of basis functions will cause a lot of computation in wavelets related method. In the meanwhile, Gaussian process prior on $g(\cdot)$ will make the Markov Chain Monte Carlo (MCMC) computationally intensive even when the sample size is not very large because calculating the inversion of $(n \times n)$ covariance matrix is required in every MCMC iteration if we set Gaussian process prior on $g(\cdot)$.

In this chapter, we propose the Bayesian Wilcoxon rank-based estimate for SIM by using the same framework introduced in Section 2.1.

4.2 Model Formulation

We consider the single-index model

$$Y_i = g\left(X_i' \boldsymbol{\beta}\right) + \varepsilon_i, \quad i = 1, \ldots, n$$

where the X_i are predictors, and the Y_i are response variables. ϵ_i are independent random errors which follows $N\left(0, \sigma^2\right)$. To ensure identifiability of the model, the unknown indexing coefficient $\boldsymbol{\beta}$ is normalized to have unit Euclidean norm and its first element restricted to be positive. The variance σ^2 is also unknown. The function $g(\cdot)$ is an unknown smooth link function.

Since modeling the link function is not the priority we are interested here, we simply apply regression splines to approximate the conditional distribution of $g(\cdot)$ given β and σ^2. The basis functions for estimation the link function $g(\cdot)$ must be computed inside JAGS because the single index depends on parameter values. We use the following truncated line model

$$g(s) = \alpha_0 + \alpha_1 s + \textstyle\sum_{k=1}^{K} u_k \left(s - \kappa_k\right)_+, \quad u_k \text{ i.i.d. } N\left(0, \sigma_u^2\right)$$

with priors $\alpha_0, \alpha_1 \overset{\text{ind}}{\sim} N\left(0, 10^8\right)$ and $\sigma_u \sim$ Half-Cauchy (25), where $s = X_i'\boldsymbol{\beta}$. To ensure the identifiability in JAGS, we use the following spherical coordinates to impose the restriction $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$. This is achieved by reparametrizing the model using the polar coordinate system

$$\beta_1 \quad = \quad r\cos\left(\varphi_1\right)$$

$$\begin{aligned}
\beta_2 &= r\sin(\varphi_1)\cos(\varphi_2) \\
\beta_3 &= r\sin(\varphi_1)\sin(\varphi_2)\cos(\varphi_3) \\
&\cdots \\
\beta_{n-1} &= r\sin(\varphi_1)\cdots\sin(\varphi_{n-2})\cos(\varphi_{n-1}) \\
\beta_n &= r\sin(\varphi_1)\cdots\sin(\varphi_{n-2})\sin(\varphi_{n-1})
\end{aligned}$$

Based on the same framework we used in Chapter 2 and 3, we set the likelihood for Bayesian Wilcoxon estimate to be

$$\mathrm{Y}_i - \mathrm{Y}_j \sim \mathit{Laplace}\left(g\left(X_i'\boldsymbol{\beta}\right) - g(X_j'\boldsymbol{\beta}), \Sigma\right)$$

As we mentioned in section 2.2, we need to figure out a way to deal with the correlation caused by pairing the data. Based on the same idea of sandwich variance, we propose a standard error correction for the SIM using working correlation matrix approach. Since our method is based on a Bayesian representation of the least absolute deviations method, we will give a brief description of the method in the next section.

4.3 Bayesian Least Absolute Deviations Method

Instead of using least-squares method, least absolute deviations (LAD) is one of the alternative methods that can provide robust estimators. We have discussed that the Bayesian Wilcoxon estimate is obtained by calculating the LAD estimator on paired data. It is not difficult to see that if the likelihood is set to be $\mathrm{Y}_i \sim \mathrm{Laplace}(g(X_i'\boldsymbol{\beta}), \Sigma)$ in JAGS, the LAD estimator based on the original data under Bayesian framework can be obtained directly. Bayesian LAD estimate may not be as efficient as Bayesian Wilcoxon estimate, but it is still robust and it does not have the covariance issue caused by pairing the data.

It is well known that if the errors follow a Laplace distribution in regression modeling, LAD is also the maximum likelihood estimate. This optimality is equivalent to that the least squares estimator has when the errors follow a normal distribution. This can be easily proved from the density function of the Laplace distribution which includes a term for the mean absolute deviation rather than the squared deviation that features in the normal distribution.

A least squares estimated line will be affected by all of the data points, while LAD regression is robust because outliers do not have a large effect on the regression line. This is also why it lacks efficiency. If there exists an outlier, since a LAD line must cross two data points, that outlier will not be one of those two points because that will not minimize the sum of absolute deviations. The least squares estimated line always adjusts itself when a data point is changed. However the LAD line may not move at all unless the added data points are central in the sense of residual distributions.

Definition 4.3.1. Let $T\left(H\right)$ be a functional defined on the set of distribution functions. We will say that $T\left(H\right)$ is a location functional if the following three conditions hold:
First, $T\left(H_{aX+b}\right) = aT\left(H_X\right) + b, a > 0$ (location-scale equivariance).
Second, $T\left(H_{-X}\right) = -T\left(H_X\right)$ (symmetry).
Third, if G is stochastically larger than F (ie.$G(x) \leq F(x)$), which means $\forall$ x, we have $T(G) \geq T(F)$ (stochastic order).
Then $\theta = T\left(H\right)$ is called a location parameter of H.

Definition 4.3.2. Let $S(\theta)$ be a gradient function and n is the sample size. We say an estimating function $S(\theta)$ is Pitman Regular if

1. $S(\theta)$ is nonincreasing against θ.

2. Assume $\bar{S}(\theta) = S(\theta)/n^{\gamma}$, $\exists$ $\gamma > 0$ and a function $\mu(\theta)$, such that $\mu(0) = 0$, $\mu'(\theta)$ is continuous at 0, $\mu'(0) > 0$ and either $E_\theta(\bar{S}(0) = \mu(\theta)$ or $\bar{S}(0) \overset{P_\theta}{\to} \mu(\theta)$;

3. $\forall$ $B > 0$,
$$\sup_{|b| \leq B} \left| \sqrt{n}\bar{S}\left(\frac{b}{\sqrt{n}}\right) - \sqrt{n}\bar{S}(0) + \mu'(0)b \right| \overset{P}{\to} 0$$

4. $\exists$ a constant $\sigma(0)$ such that
$$\sqrt{n}\left\{\frac{\bar{S}(0)}{\sigma(0)}\right\} \overset{\mathcal{D}_0}{\to} N(0,1)$$

so that

$$k = \mu'(0)/\sigma(0)$$

is called the efficacy of $S(\theta)$.

The LAD estimation method has been around for couple of centuries but rigorous methods of computing it were only introduced beginning in the 1950's and notably by Gentle in 1977 (Gentle 1977). The associated dispersion for LAD is given by

$$D_1(\theta) = \sum_{i=1}^{n} |X_i - \theta|$$

while the negative gradient function is given by

$$S_1(\theta) = \sum_{i=1}^{n} \operatorname{sgn}(X_i - \theta)$$

since the L_1 norm is defined as

$$\|\mathrm{x}\|_{L_1} = \sum_{i=1}^{n} |x_i|$$

We need to estimate the following equation

$$0 = n^{-1}\sum_{i=1}^{n} \operatorname{sgn}(x_i - \theta) = \int \operatorname{sgn}(x - \theta) dF_n(x)$$

where F_n is the empirical cumulative distribution function. It can be seen that the solution is the median of the observation. By replacing the empirical cumulative distribution function (cdf) by the true underlying cdf F, the above equation can be rewrite as

$$0 = \int \operatorname{sgn}(x - T(F)) dF(x) = -\int_{-\infty}^{T(F)} dF(x) + \int_{T(F)}^{\infty} dF(x)$$

where $T(F)$ is a location functional, and $\theta = T(F)$ is a location parameter of F. Then, we can find $T(F) = F^{-1}(1/2)$ as expected since we know that $F(T(F)) = 1/2$.

$\widehat{\theta}$ has an asymptotic $N(\theta, \tau_S^2/n)$ distribution where $\tau_s = 1/(2f(\theta))$. Assume that $f(0) > 0$, it was showed in Hettmansperger (2010) that the efficacy of the L_1 is $2f_0$ since L_1 gradient process is Pitman regular.

LAD regression does not have an analytical solving method unlike least squares regression though the ideas of these two are both straightforward, hence an iterative approach like simplex method can be applied. There are a lot of linear programming approach(including the simplex method) can be applied here because the problem is a linear program. According to William (William 2002), simplex-based methods are the preferred method. It is known that at least one LAD line cross at least two points in the data. Simplex method chooses the line by comparing the smallest absolute error over all of the data points of each line. However, there will be multiple solutions if multiple lines have the same smallest absolute error. Using Bayesian approach allows us to obtain LAD estimate through Markov chain Monte Carlo (MCMC) methods without any additional computational cost comparing with Bayesian least square estimate.

Another appealing feature of Bayesian LAD is that the scale parameter is estimated in the Bayesian interactions without having to use a preliminary fit. This makes Bayesian LAD more efficient than normal LAD method. Similarly, our Bayesian Wilcoxon rank-based estimate has the same feature. This estimated scale parameter is then used on obtaining the corrected standard errors of the model parameter as $Cov(\hat{\beta}) \approx \tau_S^{-2}\left(X^{\mathrm{T}}\mathbf{A}^{-1}X\right)^{-1}$, where $\mathbf{A}$ is the covariance structure given in Chapter 2. In Zhan (2009), they estimated the scale parameter in a frequentist perspective, and they did not integrate its estimation into the Bayesian inference machinery.

In Chapter 3, we showed that Bayesian Wilcoxon rank-based estimate is feasible in linear model with measurement error. Identifiability issues make this quite complicated for single index model with the measurement error in the predictor. As we mentioned in Chapter 3, Bayesian analysis treat the true unobserved value as a latent variable. By adding the measurement error model $V_i|X_i \sim N\left(X_i, \sigma_u^2\right)$ into the single index model derived in this chapter, one expects that it is possible to obtain a Bayesian Wilcoxon rank-based estimate for SIM with measurement error in the regressor variable. However, according to simulation studies we performed, the Bayesian Wilcoxon rank-based approach based on paired differences suffers from unidentifiability issues. This can be seen by noting that the dependence of $g\left(X_i'\boldsymbol{\beta}\right) - g\left(X_j'\boldsymbol{\beta}\right)$ on $X_i - X_j$ is not clearly discernible.

We conclude that if the model is too complicated like the SIM with measurement error case, pairing the data will cause identifiablity issues. Hence our Bayesian Wilcoxon rank-based estimate may fail. Meanwhile, the Bayesian LAD estimate is an alternative method which is also robust and is not affected by identifiabiltiy issues caused by differencing. Moreover, the computational costs associated with Bayesian LAD estimation is far lower than the Bayesian Wilcoxon estimate. This prompts us to explore this approach for more complex semiparametric models including those with measurement error in their regressor variables. That said, we still expect the Bayesian Wilcoxon estimator to provide favorable performance in the single index model without measurement error in its regressor variables.

4.4 Standard Error Correction for Single Index Model

As we mentioned in Section 2.2, $\widehat{\beta}$ is still asymptotically unbiased but it is no longer efficient with this covariance issue. We use working correlation structures to correct $Cov(\hat{\boldsymbol{\beta}})$ like what we did with linear model. Unlike the linear model case, the derivative of $E_(Y_i)$ with respect to $\boldsymbol{\beta}$ needs to be calculated before the correction. In linear model, $\nabla_{\boldsymbol{\beta}}\left(\boldsymbol{X}_i^T\boldsymbol{\beta}\right) = \boldsymbol{X}_i$, but in SIM, $\nabla_{\boldsymbol{\beta}}\left(g_{\boldsymbol{\beta}}\left(\boldsymbol{X}_i^T\boldsymbol{\beta}\right)\right) = X_i\nabla_{\left(\boldsymbol{X}_i^T\boldsymbol{\beta}\right)}\left(g\left(\boldsymbol{X}_i^T\boldsymbol{\beta}\right)\right)$. Since we used regression splines to approximate the link function g, the derivative of the link function can be obtained through JAGS directly. The approximated function is presented as n points. If the sample size is large, we can use $\left(g\left(\boldsymbol{X}_{i+1}^T\boldsymbol{\beta}\right) - g\left(\boldsymbol{X}_i^T\boldsymbol{\beta}\right)\right) / \left(\boldsymbol{X}_{i+1}^T\boldsymbol{\beta} - \boldsymbol{X}_i^T\boldsymbol{\beta}\right)$ to calculate the derivative. To be more accurate, the following truncated model is used to calculate the derivative because the derivatives of spline functions can be simply expressed in terms of lower order spline functions.

$$g(s) = \alpha_0 + \alpha_1 s + \sum_{k=1}^{K-1} u_k^{'}\left(s - \kappa_{k-1}\right)_+, \quad u_k \quad i.i.d. \quad N\left(0, \sigma_u^2\right)$$

where $u_k^{'} = (u_{k+1} - u_k)/(\kappa_{k+1} - \kappa_k)$. The covariance matrix can be obtained in the following form

$$\Sigma = E\left\{ I_{\Gamma}(\mathbf{X_i})\nabla_{\boldsymbol{\beta}}\left(g_{\boldsymbol{\beta}}\left(\mathbf{X}_i^T\boldsymbol{\beta}\right)\right)\left[\nabla_{\boldsymbol{\beta}}\left(g_{\boldsymbol{\beta}}\left(\mathbf{X}_i^T\boldsymbol{\beta}\right)\right)\right]^T\right\}$$

where $I_{\Gamma}(\mathbf{X_i})$ is trimming device to keep the estimator away from zero. Thus, $Cov(\hat{\boldsymbol{\beta}}) = \left(\left(g_{\boldsymbol{\beta}}\left(\mathbf{X}_i^T\boldsymbol{\beta}\right)\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(g_{\boldsymbol{\beta}}\left(\mathbf{X}_i^T\boldsymbol{\beta}\right)\right)\right)^{-1}$.

4.5 Simulation and Real Data Example

4.5.1 Simulation

We compare three underlying models to show the finite samples performance among the least squares (LS), the least absolute deviation (LAD), and our rank (Rank) estimators under Bayesian Framework.

Case 1: $Y = 1.5\sin\left(\left(\beta_0'\mathbf{X}\right)\pi\right) + \varepsilon, \quad \beta_0 = (1,2,0,2)/3, \mathbf{X} \sim (U[-1,1])^{\otimes 4}$
Case 2: $Y = 4\sqrt{|(\beta_0'\mathbf{X}) + 1|} + (\beta_0'\mathbf{X}) + \varepsilon, \quad \beta_0 = (2,-2,4,-1)/5, \mathbf{X} \sim (U[-2,2])^{\otimes 4}$
Case 3: $Y = 2\left(\beta_0'\mathbf{X}\right) + 10\exp\left(-\left(\beta_0'\mathbf{X}\right)^2/5\right) + \varepsilon, \quad \beta_0 = (2,-2,4,-1)/5, \mathbf{X} \sim (U[-2,2])^{\otimes 4}$

As an oscillating function, Case 1 is commonly used in SIM study (Liu et al. , 2013). Case 2 and 3 were first studied by Zeng et al. (2012). For case 2, the function $g(\cdot)$ is a non-differentiable function with a corner point which is usually difficult to capture. The model errors are generated from three different distributions to study the robustness of the proposed estimator. We used standard normal distribution, t-distribution with degrees of freedom 3, and contaminated normal distribution. The t-distribution with degrees of freedom 3 has a thick tail and contaminated normal distribution contains a few extreme large outliers (about 5 %). The setting of this simulation is similar to the simulation in Abebe et al. (in press).

In table 4.1, we report the angle between the true β and the estimated β. We use MSE to describe the performance of estimating $g(\cdot)$. Figure 4.1 shows the estimated functions for all three cases.

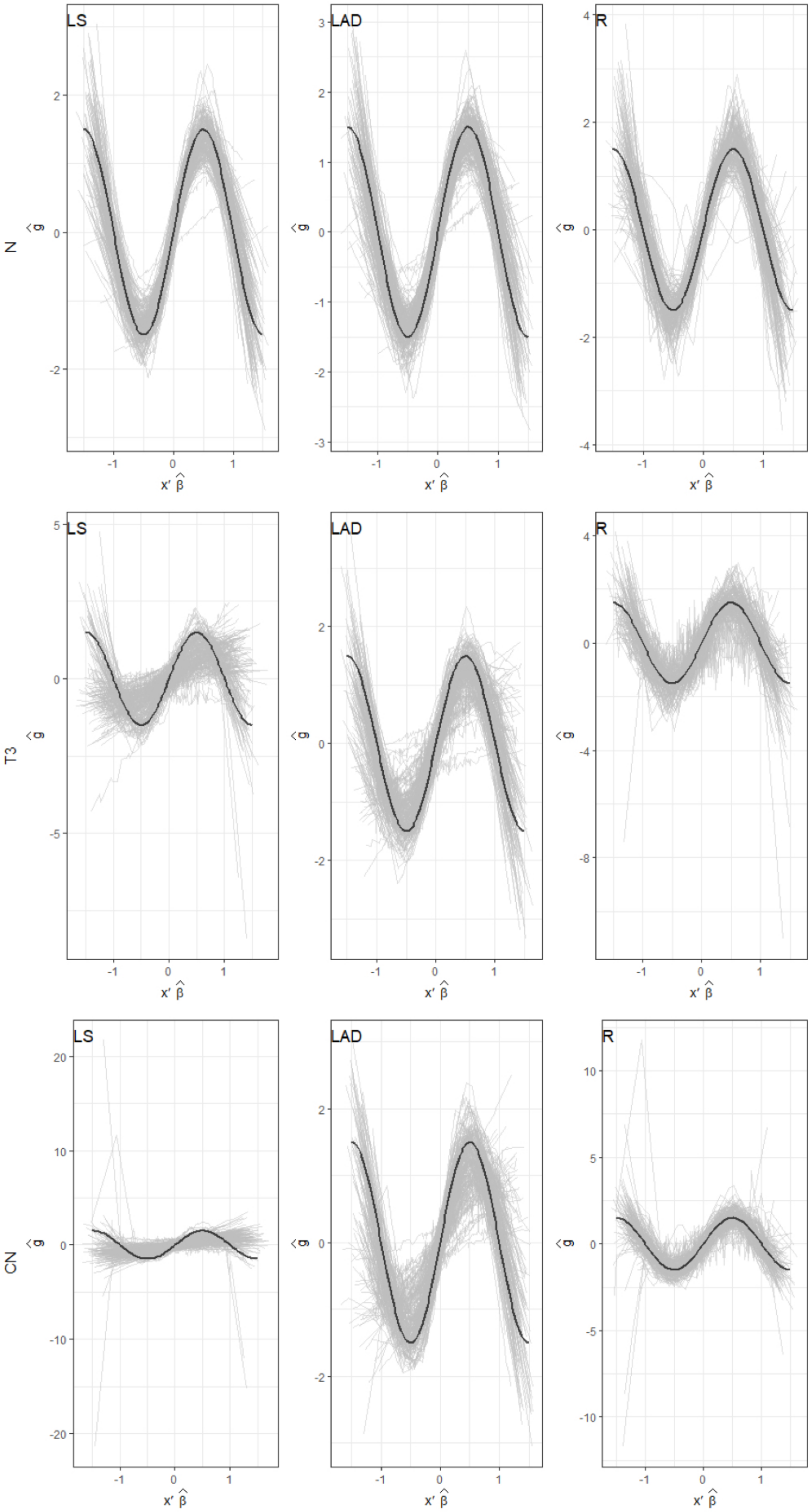

Figure 4.1: Estimated functions for Case 1

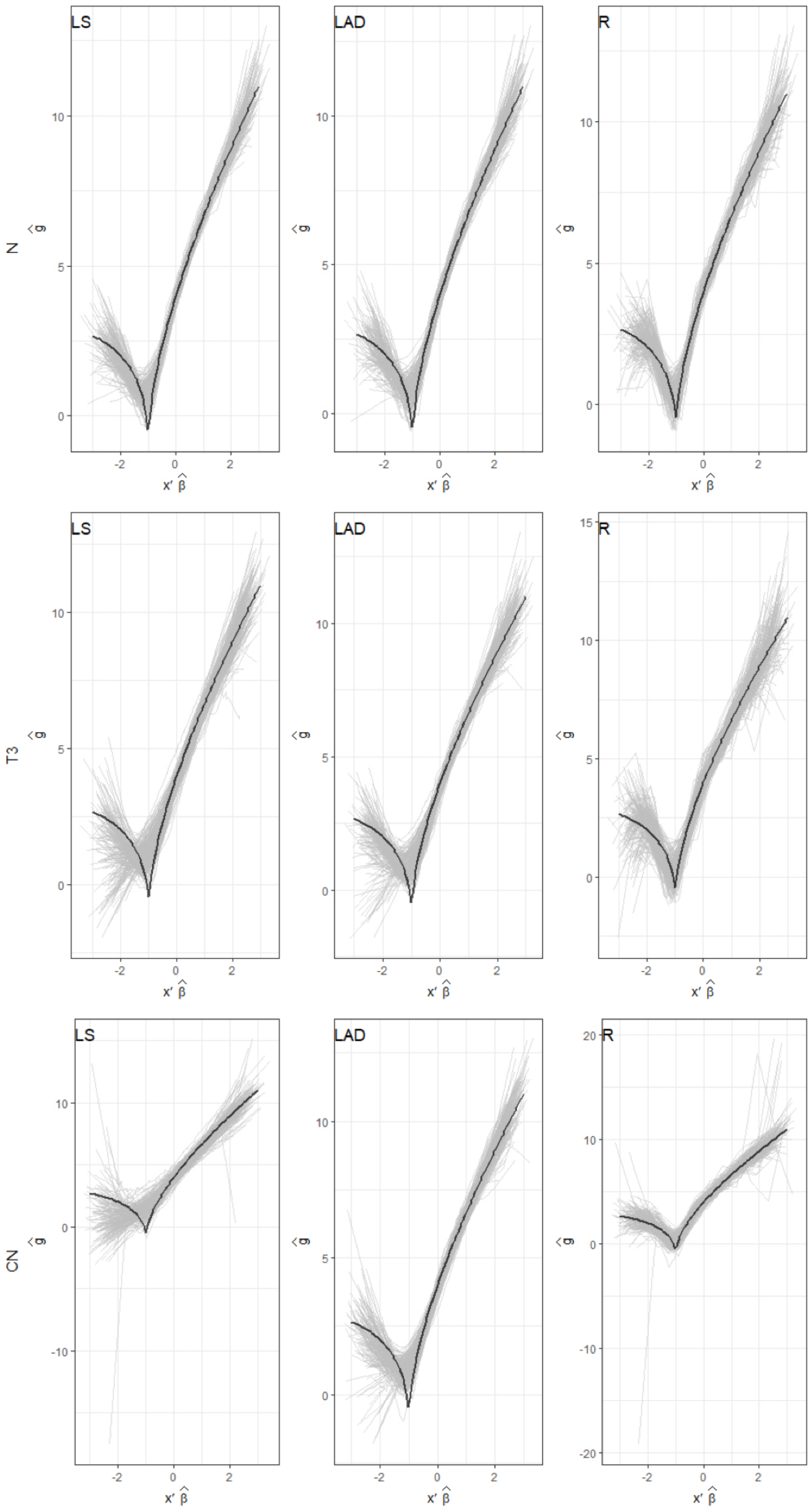

Figure 4.2: Estimated functions for Case 2

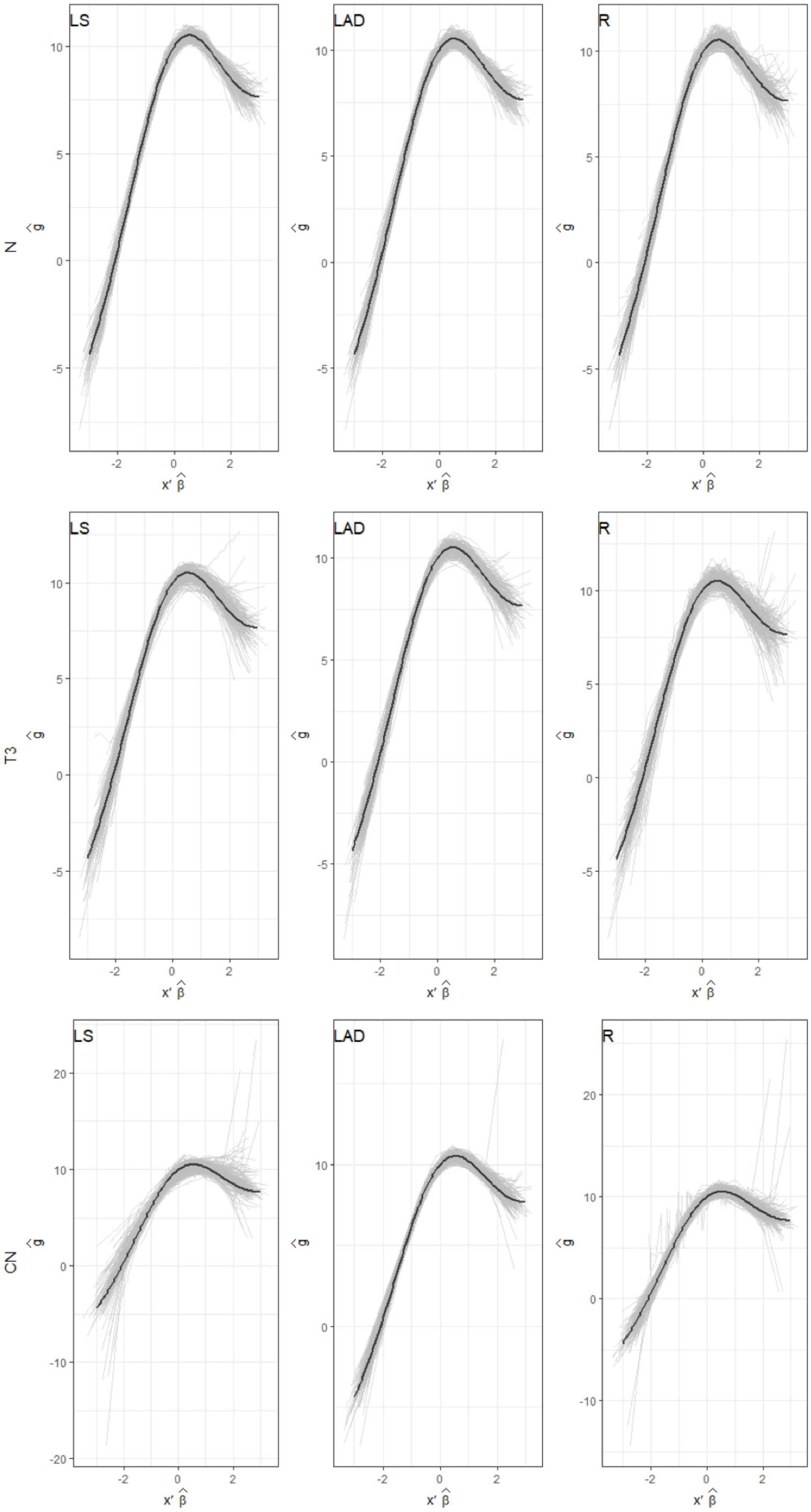

Figure 4.3: Estimated functions for Case 3

	ϵ	BLS			BLAD			BRank		
		Mean	SD	MSE(g)	Mean	SD	MSE(g)	Mean	SD	MSE(g)
Case 1	$N(0,1)$	5.96	2.85	7.79	6.64	4.19	9.32	6.37	6.54	12.15
	t_3	17.80	10.22	48.69	9.18	6.27	18.46	8.59	5.97	22.94
	CN	25.15	10.60	79.55	9.39	6.55	19.70	7.47	6.57	23.74
Case 2	$N(0,1)$	3.00	1.29	13.83	3.41	1.52	14.97	3.33	1.46	13.76
	t_3	5.36	5.29	36.90	4.13	1.98	24.57	4.31	2.29	23.26
	CN	8.14	4.18	67.49	3.72	1.74	21.88	3.85	1.83	28.05
Case 3	$N(0,1)$	2.71	1.20	6.20	2.91	1.33	7.65	2.84	1.25	9.76
	t_3	4.09	2.20	15.83	3.34	1.59	10.40	3.57	1.64	18.17
	CN	6.42	4.14	41.65	2.98	1.38	8.28	3.19	2.11	22.84

Table 4.1: The Mean and SD of the angle and the MSE of the function for all Cases

The Bayesian LS (BLS) method only gives better performance than the Bayesian LAD (BLAD) and the Bayesian Wilcoxon rank (BRank) methods in all three cases for normal errors and the BRank method provides better performance than BLAD under normal errors as expected. For Case 1, when considering the t_3 and the contaminated normal error distribution, BRank method provides a better estimator on β, but BLAD method has a better performance on estimating the function $g(\cdot)$. In Case 2, BLAD and the BRank methods provide similar results under t_3 and the contaminated normal error distribution, but BLAD estimator performs better than BRank estimator in Case 3. In terms of modeling the function $g(\cdot)$, the BLAD method performs as well as the BRank method. BLAD method has the least MSE in many scenarios regarding the function $g(\cdot)$, but Bayesian rank method shows the best ability to capture the shape of the function in the middle part (Figure 4.1). Especially in case 2, BRank method fit much better around the corner point than the others. Comparing the result with the frequentist approach given in Abebe *et al.* (in press), all of these three methods perform better under the proposed Bayesian framework.

4.5.2 Real Data Example

In this section, we use the Boston Housing Dataset, which has been used extensively throughout the literature to demonstrate the performance of our method. This dataset reports the median value of owner-occupied homes in 506 U.S. census tracts in the Boston area, together with 14 variables which may be useful to explain the variation in median value of owner-occupied homes in USD 1000's. In our study, we only use rm, $\log(tax)$, $ptratio$ and $\log(lstat)$ as the variables. rm means average number of rooms per dwelling, tax is full-value property-tax rate per \$10,000, $ptratio$ represents student-teacher ratio by town, and $lstat$ indicates lower status of the population (percent).

Similar to the previous simulation, we compare the least squares (BLS), the least absolute deviation (BLAD), and our rank (BRank) estimators under the Bayesian framework in a cross-validation study. We randomly select 20% of the data to be test data and the remaining 80% as the training data. The cross-validation results are as follows:

	MSE	Variance
BLS	143.2768	116.1255
BLAD	146.9098	103.5769
BRank	145.6380	104.1171

Table 4.2: Cross validation result of Boston housing Data

According to the MSE, our method provides a close result to Bayesian least squares estimator. Like the Bayesian LAD estimator, the Bayesian rank estimator is quite robust.

	Sample Size: 300		
	BLS	BLAD	BRank
rm	0.692 (0.048)	0.722 (0.036)	0.682 (0.003)
$\log(tax)$	-0.495 (0.068)	-0.422 (0.070)	-0.519 (0.005)
$ptratio$	-0.069 (0.008)	-0.068 (0.010)	-0.069 (0.001)
$\log(lstat)$	-0.513 (0.045)	-0.536 (0.048)	-0.511 (0.003)
MSE	7.456	7.525	7.383
	Sample Size: 500		
rm	0.424 (0.049)	0.573 (0.039)	0.516 (0.002)
$\log(tax)$	-0.283 (0.081)	-0.474 (0.051)	-0.498 (0.003)
$ptratio$	-0.075 (0.012)	-0.080 (0.009)	-0.077 (0.001)
$\log(lstat)$	-0.850 (0.041)	-0.659 (0.044)	-0.692 (0.003)
MSE	19.312	20.287	19.773

Table 4.3: Fitting result of Boston Housing Data with different sample sizes

Table 4.3 shows the fitting results based on the same data using different sample sizes. When we use the whole data set, BLS estimator is the best and BLAD estimator is the worst as expected. After decreasing the sample size from 500 to 300, BRank method performs best among these three. The reason is that when sample size is relatively small, outliers will affect the result heavily. Figure 4.4 is a QQ-plot of the residuals. The right tail for the BRank estimate is closer to the expected line compared to the BLS estimate which shows the robustness of the BRank estimate. Comparing with the BLAD estimate, the BRank estimate is closer to the expected line on the left tail and the main part in the middle which indicates that the BRank estimate is more efficient than the BLAD estimate.

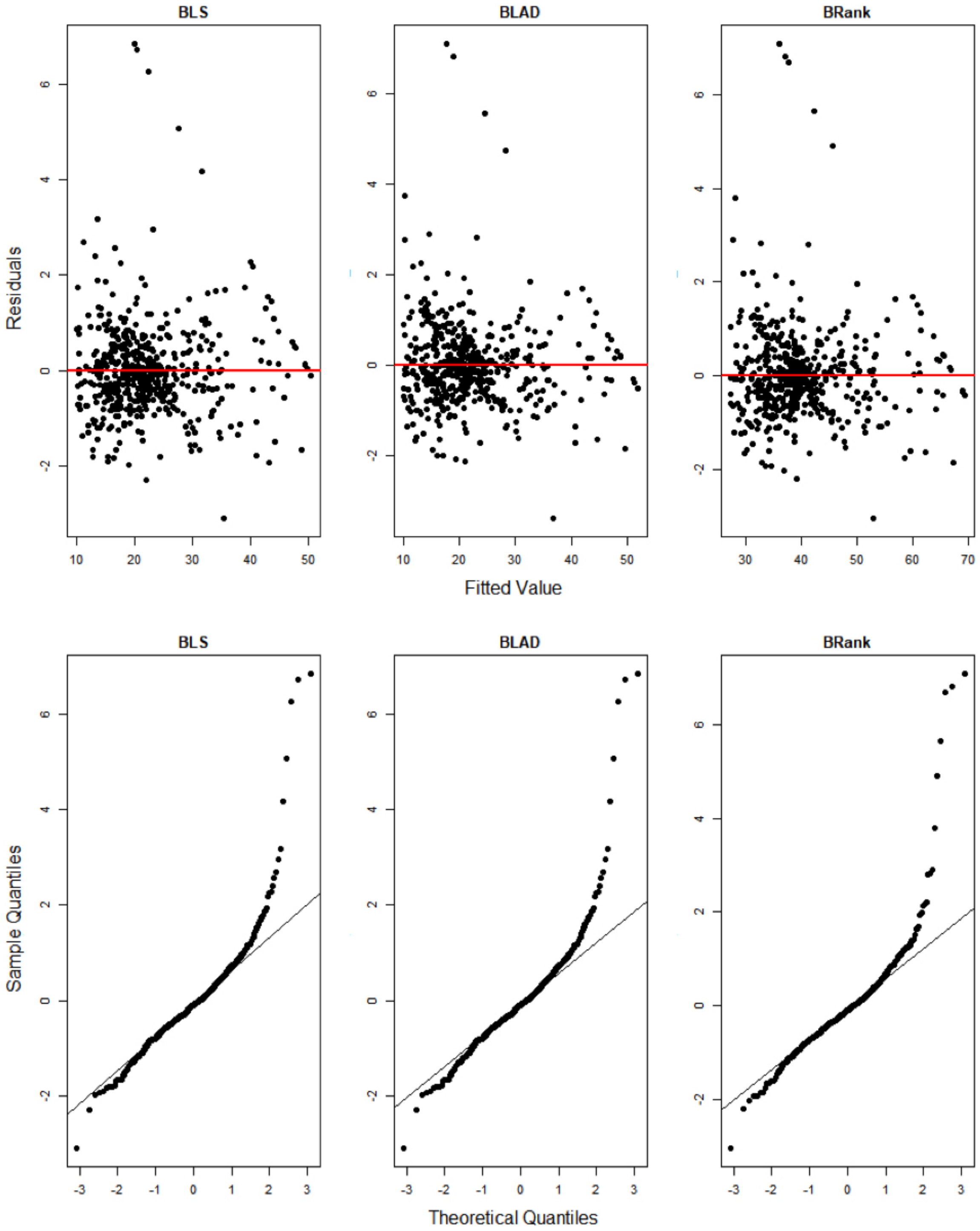

Figure 4.4: QQ-plot of the residuals

Chapter 5

Single-Index Varying Coefficient Models

5.1 Varying Coefficient Model

To deal with the "curse of dimensionality" in high-dimensional data, many powerful approaches have been developed. One of those approaches, the varying coefficient (VC) model, is an extension of simple linear models in which the regression coefficients are taken to be unknown smooth functions that change as a function of another variable. Suppose Y_i is the response variable, $\boldsymbol{X}_i = (\mathrm{x}_{0i}, \ldots, \mathrm{x}_{pi})'$, and $U_i = (u_{1i}, \ldots, u_{qi})'$ are predictor variables. In order to allow a varying intercept term in the model, we set $\mathrm{x}_{0i} = 1$. The vary coefficient model (VCM) is defined as

$$Y_i = \{\boldsymbol{g}\,(U_i)\}'\,\boldsymbol{X}_i + \varepsilon_i \quad i = 1, \ldots, n$$

where $\boldsymbol{g}(\cdot) = (g_0(\cdot), \ldots, g_p(\cdot))'$ is a p-vector of unknown coefficient functions.

The "curse of dimensionality" can be avoided in varying coefficient models because the unknown coefficient functions $\boldsymbol{g}(\cdot) = (g_0(\cdot), \ldots, g_p(\cdot))'$ are dependent on U. Interpretability is the other big advantage of using the varying coefficient model. When someone is trying to explore the situation that the regression coefficients change over time or other variables, this model will be more flexible.

We use the same approach we used in the case of single index models to capture the unknown function $g(\cdot)$, and we set the likelihood in JAGS to be

$$\mathrm{Y}_i \sim \text{Laplace}\left(\{\boldsymbol{g}\,(U_i)\}'\,\boldsymbol{X}_i, \Sigma\right)$$

so that the BLAD estimate can be obtained directly.

Similarly, by setting the likelihood to be

$$\mathrm{Y}_i - \mathrm{Y}_j \sim \text{Laplace}\left(\{\boldsymbol{g}\left(U_i\right)\}' \boldsymbol{X}_i - \{\boldsymbol{g}\left(U_j\right)\}' \boldsymbol{X}_j, \Sigma\right)$$

we can obtain the Bayesian Wilcoxon rank-based estimate.

5.2 Single Index Varying Coefficient Models

Because of its interpretability and flexibility, the single index varying coefficient model (SIVCM) is studied by a lot of researchers in public health, ecology and other fields. The SIVCM model has a similar set up as the VC model but it allows for the coefficient functions to depend on high dimensional variables using SIM type projection pursuit. Since it is a combination of SIM and VC model, SIVCM inherits the ability to overcome the "curse of dimensionality" from VC model. This makes SIVCM vary attractive to those who are using nonparametric models with multivariate data. The SIVCM has the following form:

$$y_i = \left\{\boldsymbol{g}\left(\boldsymbol{\beta}_0^T \boldsymbol{Z}_i\right)\right\}^T \boldsymbol{X}_i + \varepsilon_i \quad i = 1, \ldots, n$$

where $\boldsymbol{\beta}_0$ is a vector of unknown regression parameters representing the single-index part, $\boldsymbol{g}(\cdot) = (g_0(\cdot), \ldots, g_p(\cdot))'$ is a unknown coefficient functions and are random errors. To ensure the identifiability, we assume that $\|\boldsymbol{\beta}_0\| = 1$ and $\beta_{01} > 0$. SIVCM can be reduced to VC model if the dimension of the regression parameters is 1. Similarly, it can be reduced to SIM as well if the dimension of the coefficient functions is 1. There are many existing approaches for SIVCM based on least squares methods.

The estimator of $\boldsymbol{\beta}_0$ in SIVCM based on least squares method was first introduced by Xia and Li (1999). They proved that their proposed estimator was $\sqrt{n}$-consistent and asymptotically normally distributed under some regularity conditions. Another estimate based on a profile least squares local linear regression was proposed by Fan et al. (2003). It is more computationally efficient because they selected locally significant variables based on the Akaike information criterion (AIC). Xue and Pang (2013) proposed the estimations of both the regression parameters and the coefficient functions using the "remove-one-component" idea.

For the purpose of dealing with outliers, model error with heavy-tail distributions, and model contamination, some robust estimation procedures were proposed recently. Yao et al. (2012) introduced a local modal estimation procedure using EM algorithm for nonparametric regression models. They showed that their estimator was asymptotically as efficient as other least squares based estimators if the model errors follow the normal distribution or there are no outliers. The idea of using local modal estimation has been extended to VC model, SIM, and SIVCMs. All of the approaches were shown to have some advantages like robustness.

Sun (2017) and Sun et al. (2019) proposed a rank-based estimation procedure for SIVCM. He used local linear estimation for estimating the coefficient functions with bandwidth selection, and a backfitting type algorithm for computing the regression coefficient. The computational cost is relevantly expensive since the procedure heavily relied on cross-validation.

Like we mentioned in Section 5.1, in order to get the BLAD estimate for SIVCM, we need to set the likelihood to be

$$\mathrm{Y}_i \sim \text{Laplace}\left(\left\{\boldsymbol{g}\left(\boldsymbol{\beta}_0^T \boldsymbol{Z}_i\right)\right\}' \boldsymbol{X}_i, \Sigma\right)$$

Similarly, by setting the likelihood to be

$$\mathrm{Y}_i - \mathrm{Y}_j \sim \text{Laplace}\left(\left\{\boldsymbol{g}\left(\boldsymbol{\beta}_0^T \boldsymbol{Z}_i\right)\right\}' \boldsymbol{X}_i - \left\{\boldsymbol{g}\left(\boldsymbol{\beta}_0^T \boldsymbol{Z}_i\right)\right\}' \boldsymbol{X}_j, \Sigma\right)$$

we can obtain the Bayesian Wilcoxon estimate. Unlike the SIM with measurement error case, Bayesian Wilcoxon estimate is not unfeasible in SIVCM but it is not efficient. To the best of our knowledge, a large size of sample will be needed in order to get a result as expected for a simple SIVCM with low dimensions. For example, in order to guarantee a proper result on a SIVCM with 2 regression parameters and 2 coefficient functions, the sample size has to be at least 200. More discussion can be found in section 5.4.

5.3 Simulation for VC Model

In this simulation, we consider the easiest VC model:

$$Y_i = g_0(u_i) + g_1(u_i) X_{1i} + \varepsilon_i$$

where we set $g_0(u)$ to be $1 + 3u^2$, and $g_1(u)$ to be $3\exp(-u^2)$. We used 150 as the sample size with 100 simulation runs to show the performance of the same three methods (BLS, BLAD and BRank). The number of knots we use is determined by cross-validation to minimize the prediction errors. Since we want to show the robustness of BLAD and Bayesian Wilcoxon rank-based method (BRank), we created a mixed normal distribution for the model error in the same manner as what we did in Section 3.4.2. Recall that this distribution is a mixture of two normal distributions where 95% of the model errors were generated from N(0, 1) and 5% of them were generated from N(0, 10).

	BLS	BLAD	BRank
$N(0,1)$	0.92 (0.0109)	0.93 (0.0113)	0.89 (0.0103)
t_3	2.97 (7.94)	2.99 (7.76)	2.86 (7.64)
CN	5.87 (9.42)	5.81 (9.09)	5.64 (8.67)

Table 5.1: Mean and Variance of MSE of three methods

In Table 5.1, we calculated the mean and the variance of the MSEs of the estimated functions to compare the performances. Since modeling the functions $g(\cdot)$ is the only thing which will affect the MSE in VC model, we can see that BRank method provides the best result in all three different situations. When the model error distribution is standard normal, all the results are close to each other. However, BRank method preformed much better than the other two methods in the presence of outliers or model errors follows a heavy-tailed distribution.

Figure 5.1 shows the estimated functions for all three cases. Based on Figure 5.1, we can clearly see that BLS method is not robust because it can be easily affected by a thick tailed error distribution like t_3, and some large value outliers. BLAD has better result than BLS method since most of the estimated curves have similar shapes, but the variation is very large. That explains why BLAD result is only slightly better than BLS result in contaminated normal error distribution setting. The estimated curves provided by BRank were not as smooth, but we can see that the estimations are relevantly consistent unlike BLS estimation and the variations are the smallest among all these three methods. For

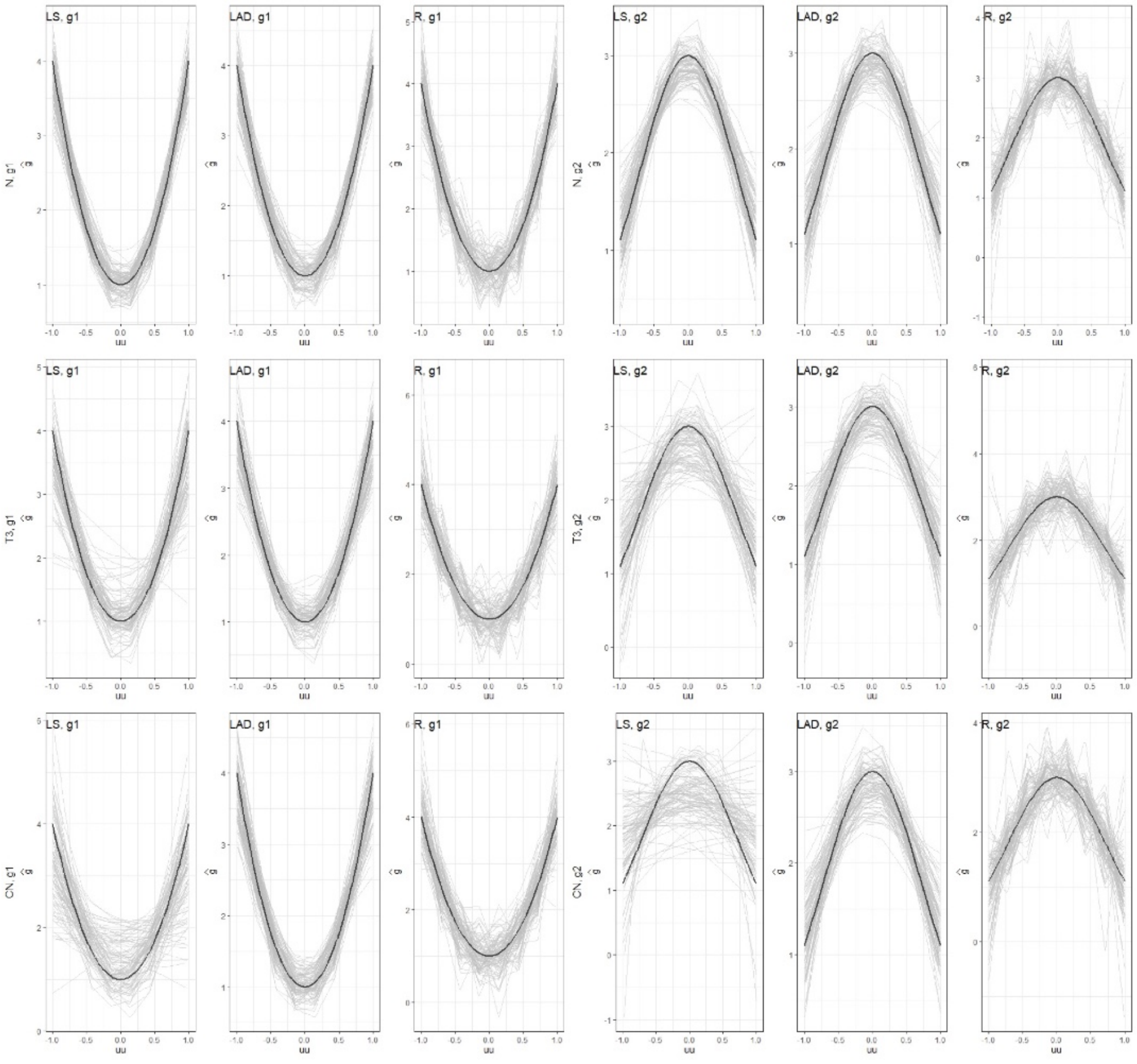

Figure 5.1: Estimated g_0 and g_1 for 3 error distributions

this specific simulation setting, 150 is the smallest sample size which can guarantee the convergence of BRank estimate. We infer that if the dimension of the model increases, we will need more data to perform BRank estimate.

5.4 Real Data Example for SIVCM

In this section, we consider a fisheries data from the Gulf of Alaska which were obtained to study the interactions between groundfish predator species. Sun (2017) originally did a research on this data. There are three species in the data, Pacific halibut, Pacific cod and sablefish. They chose the response to be the CPUE of Pacific halibut, where CPUE is determined for each species based on a catch rate defined by geographical area scale. They were interested in discovering prime attraction among these three predators while also taking into account the role of Pacific halibut as a predator on Pacific cod and sablefish. There are seven environmental variables which are supposed to have an impact on the response for all of these three species as well. They are wind direction, wind speed, significant wave height, dominant wave period, average wave period, sea level pressure and sea surface temperature. According to Sun (2017) and Sun et al. (2019), Pacific cod and sablefish's relationship with Pacific halibut depends on the environment because they prefer different habitats. Plus, two outliers were identified in the CPUE of Pacific halibut, while the sample size was 52. These are the main reasons that we should perform a BLAD estimation on the following SIVCM

$$y_i = g_0\left(\beta^T Z\right) + g_1\left(\beta^T Z\right)\mathbf{x}_{1i} + g_2\left(\beta^T Z\right)\mathrm{x}_{2i} + \varepsilon_i$$

where y_i is the CPUE of Pacific halibut, x_1 is the CPUE of pacific cod, x_2 is the CPUE of sablefish and the matrix Z contains the environmental variables.

In this section, we only demonstrated a comparison between BLS estimation and BLAD estimation using the same SIVCM. As we mentioned in section 5.2, our BRank method is not efficient in this case since the sample size is too small which can cause several problems like huge bias and serious uncertainty. Even if the BRank estimate is obtained, the process of fixing the problem caused by the covariance after pairing the data in SIVCM is more complicated than SIM case.

We performed a 10-fold cross-validation to compare the BLS estimate with the BLAD estimate. In Table 5.2, BLAD estimate provided a result with smaller prediction error. Among these 10 folds (F1 to F10), F1 and F7 caught our attention. In F7, the prediction errors were all very large since both of the outliers were in the validation set. BLS estimate performed slightly better than BLAD estimate as expected. In F1, only one of the outliers was in the validation set. Even if BLAD estimate did not get affected by the outlier in the training set, the prediction error was seriously affected by the outlier in the validation set. Since these two outliers are close to each other, the prediction error for BLS estimate was not very large. Figure 5.2 is a QQ-plot of the residials. According to this QQ-plot, we can clearly see the two apparent outliers mentioned before. Unlike the BLAD result, from the top part of Figure 5.2 we can see that the residuals for those two outliers in BLS result are smaller than the residuals in BLAD result, which means the line fitted by BLS was significantly pulled towards those two outliers. This makes the BLS analysis inefficient.

The estimated coefficient functions were shown in Figure 5.3. Comparing to the result in Sun et al. (2019), all of these estimated coefficient functions are close to the expected shapes. Even if we can not tell whether the estimation of the functions are close to the true functions or not (because they are unknown), it is not difficult to find out that the estimated coefficient functions provided by BLAD method are better than the result using BLS method. The estimated values of β are shown in Table 5.3. The absolute values of $\hat{\beta}_4$ and $\hat{\beta}_5$ are much larger than the rest, and we can tell that some of the regression parameters are not significant, but the variable selection topic is not our concern in this dissertation. From this angle, BLAD estimate catches the regression parameters better than BLS estimate. This is one of the reasons that why BLAD method is able to estimate the coefficient functions better.

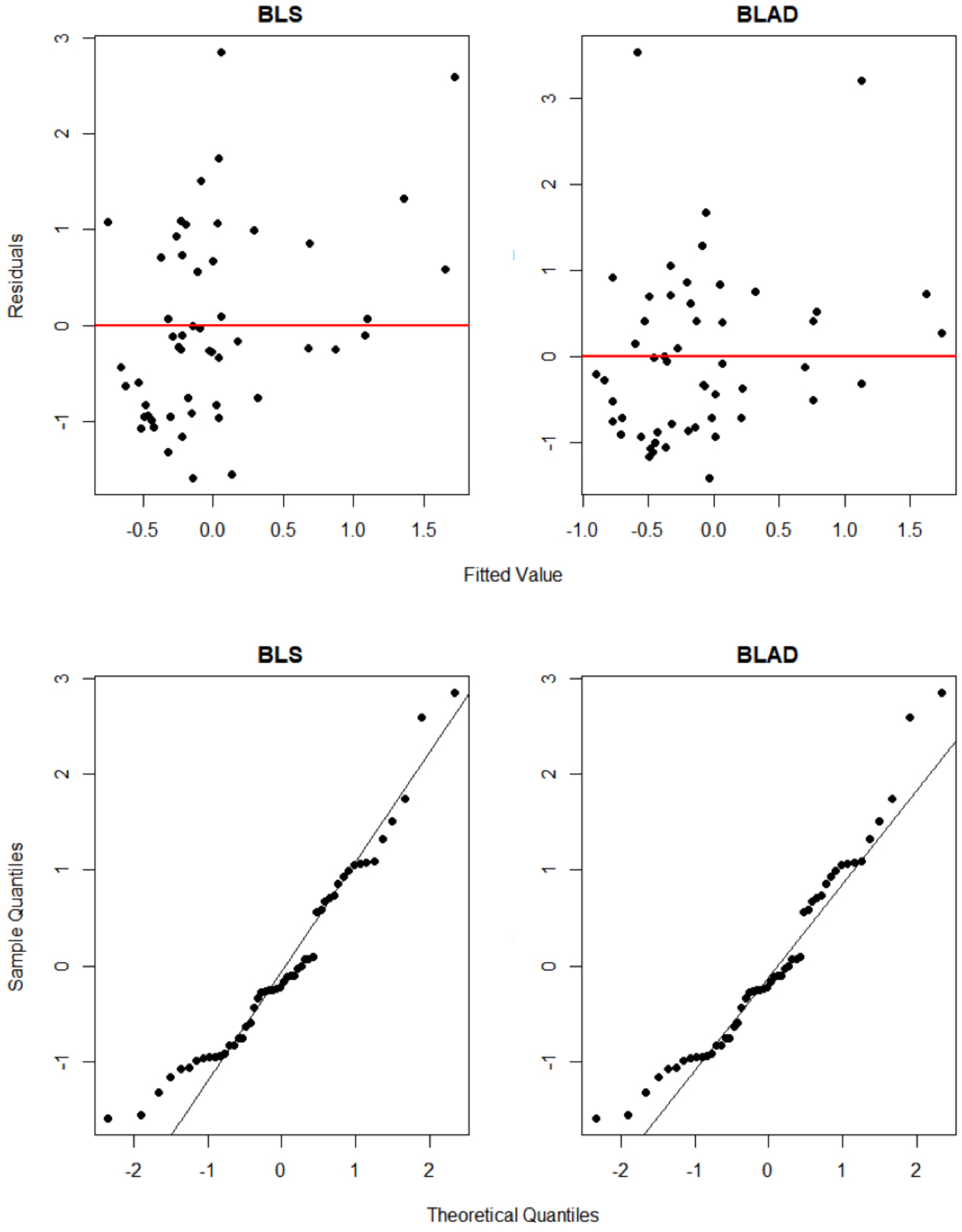

Figure 5.2: QQ-plot of the residuals

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean
BLS	0.668	0.417	0.593	0.527	1.201	0.768	3.202	0.649	1.003	1.412	1.044
BLAD	1.542	0.486	0.320	0.627	0.610	0.561	3.674	0.535	0.697	0.592	0.964

Table 5.2: 10-fold cross-validation result based on prediction errors

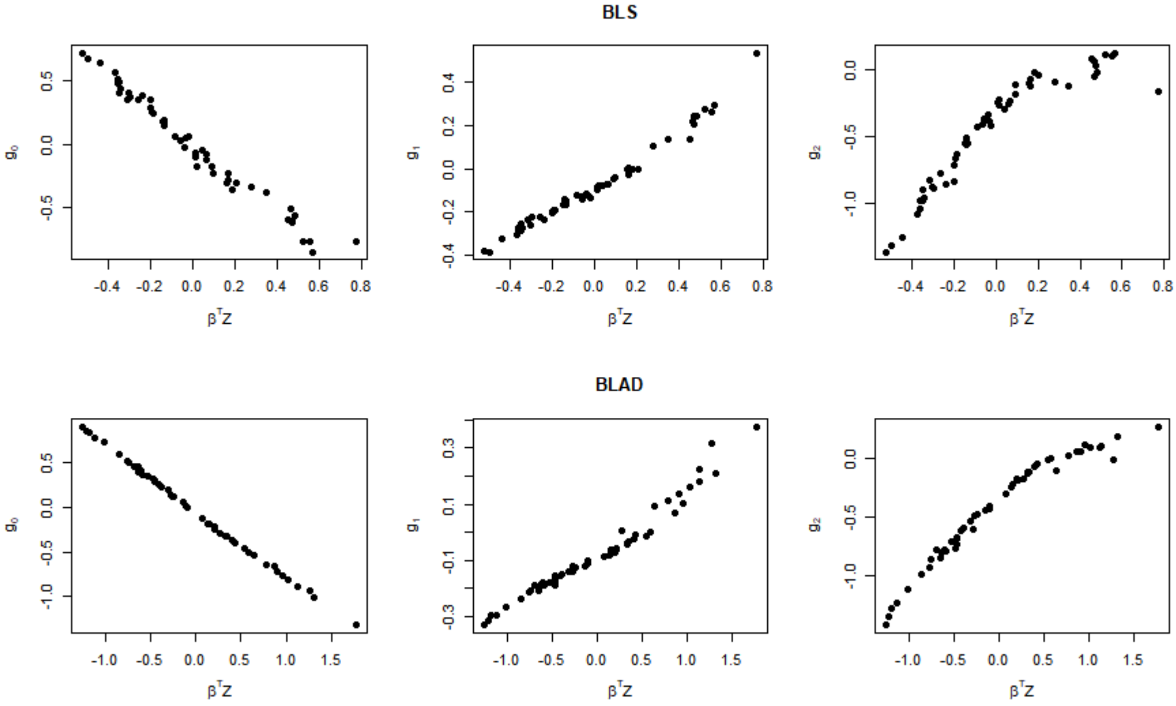

Figure 5.3: Estimated g_0, g_1 and g_2

$\hat{\beta}$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$
BLS	-0.011	0.090	-0.026	-0.180	0.282	0.047	-0.011
BRank	-0.047	0.202	0.054	-0.566	0.494	0.018	-0.056

Table 5.3: Mean of the estimated β

Chapter 6

Discussion

The purpose of this work was to provide an adaptable Bayesian approach for rank-based estimation. This has the advantage of estimating the uncertainty within the Bayesian framework avoiding a complicated density fitting approach used for estimating scale parameters in rank-regression. We also wanted the approach to be independent of the model specification so that it is flexible enough to apply for various types of model specifications including semi- and non-parametric models.

An application of this newly proposed approach for linear models with measurement error showed some favorable results in cases where the error distribution is contaminated or heavy-tailed. This is in comparison to the classic methods of handling measurement error like the SIMEX and moment-based error correction.

While our Bayesian Wilcoxon rank-based estimator has several attractive properties, there are some important issues that need to be further studied. For example, we used regression splines to approximate the link function in SIM and SIVCM. This is known to be inefficient. We made this choice because this was not a priority for our study and there is vast literature on this topic. Another big issue is that the computation cost of our Bayesian Wilcoxon rank-based estimate is too high since pairing the data makes the sample size increase to approximately $O(n^2)$. Dealing with the standard error correction issue coursed by pairing the data will also increase this cost. There is certainly a need to devise a new Bayesian sampling scheme to make the proposed approach practical for a wide range of problems.

As an alternative choice, the BLAD method provides similar result as Bayesian Wilcoxon rank-based estimate. To our knowledge, this is also the first application of this approach for semiparametric model estimation. There is always a trade off among many attractive properties, and we have explored that BLAD estimate is a robust Bayesian method which is feasible for a wide range of problems and relevantly efficient on many models. Combining Bayesian inference with other robust nonparametric methods brings advantages from both parts. However, BLAD estimate does not perform well if the model is not contaminated. This problem is inherited from LAD method, and it was not solved by combining with Bayesian inference. In practice, we think this problem may be relieved by utilizing better prior information.

References

[1] Abebe, A. F.Bindele,H., Otlaadisa, M., & Makubate, B., “Robust Estimation of Single Index Models with Responses Missing at Random,” Statistcal Papers, in press.

[2] Antoniadis, Anestis , Gregoire, Gerard & McKeague, Ian W., “Bayesian Estimation in Single-Index Models,” Universite Joseph Fourier and Columbia University, Statistica Sinica 14, 1147-1164, 2004.

[3] Bartlett, Jonathan W & Keogh, Ruth H., “Bayesian correction for covariate measurement error: A frequentist evaluation and comparison with regression calibration,” Statistical Methods in Medical Research 2018, Vol. 27(6) 1695–1708, 2016.

[4] Berkson, Joseph M.D., D.Sc., “Are there Two Regressions?,” Journal of the American Statistical Association, Volume 45, Issue 250, 1950.

[5] Buonaccorsi, John, “Measurement Error: Models, Methods and Applications,” University of Massachusetts Amherst, 2010.

[6] Carroll, Raymond J. *et al.*, “Measurement Error in Nonlinear Models, A Modern Perspective,” Second Edition, Monographs and Statistics and Applied Probability, 2006.

[7] Choi, Taeryon, “A Gaussian Process Regression Approach to Single-Index Model,” Journal of the Nonparametric Statistics, Vol.23, Issue 1, 2011.

[8] Fan, Jianqing, Yao, Qiwei & Cai, Zongwu, "Adaptive varying-coefficient linear models," Journal of the Royal Statistical Society:Series B (Statistical Methodology), 65(1), 57-80, 2003

[9] Fan, Jianqing & Zhang, Wenyang, "Statistical methods with varying coefficient models," Statistics and its Interface, Vol.1, 179-195, 2008.

[10] Ferguson, Thomas S., "A Bayesian Analysis of Some Nonparametric Problems," The Annals of Statistics, Vol.1, No.2, 209-230, University of California, Los Angeles, 1973

[11] Friedman, Jerome H. & Stuetzle, Werner, "Projection Pursuit Regression," Journal of the American Statistical Association, Vol.76, Issue 376, 1981.

[12] Geman, Stuart & Geman, Donald, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 6, November, 1984.

[13] Gentle, J, E, "Least absolute values estimation: An introduction," Commun. Stat. Simul. Comput. B6(4): 313-328, 1977

[14] Hardle, Wolfgang, Hall, Peter & Ichimura, Hidehiko, "Optical Smoothing in Single-Index Models,"Universite Catholique de Louvain, Australian National University and University of Minnesota, The Annals of Statistics, Vol.21, No.1, 157-178, 1993.

[15] Huber, P. J., "The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. I, pp. 221-33, 1967.

[16] Jureckova, Jana *et al.*, "Behavior of R-estimators under measurement errors," Bernoulli, Volume 22, Number 2, 1093-1112, 2016.

[17] Lederer, Wolfgang & Küchenhoff, Helmut, "MSIMEX- And MCSIMEX-Algorithm for Measurement Error Models ," R Package `http://wolfganglederer.github.io/simex/`, 2019.

[18] Liu, Jicai, Zhang, Riquan, Zhao, Weihua & Lv, Yazhao, “A robust and efficient estimation method for single index models,” Journal of the Multivariate Analysis, 122, 226-238, 2013.

[19] Park, Chun Gun, Vannucci, Marina & Hart, Jeffery D., “Bayesian Methods for Wavelet Series in Single-Index Models,” Journal of the Computational and Graphical Statistics, Vol.14, Issue 4, 2005.

[20] Plummer, Martyn, “JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling,” Distributed Statistical Computing March 20-22 Vienna, Austria, ISSN 1609-395X, 2003.

[21] Powell, James L., Stock, James H. & Stoker, Thomas M., “Semiparametric Estimation of Index Coefficients,” Econometrica, Vol. 57, No. 6, 1403-1430, 1989.

[22] Rousseeuw, Peter J. & Leroy, Annick M., “Robust Regression and Outlier Detection,” John Wiley & Sons, 1987.

[23] Sinay, Marick S. & Hsu, John S. J., ”Bayesian Inference of a Multivariate Regression Model,” Hindawi, Journal of Probability and Statistics, Volume 2014, Article ID 673657, 2014

[24] Stefanski, L.A., “Measurement Error Models,” Journal of the American Statistical Association, Volume 95, 2000.

[25] Stefanski, L.A. & Cook, J.R., “Simulation- Extrapolation Estimation in Parametric Measurements Error Models,” Journal of the American Statistical Association, Volume 89, Issue 428, 1994.

[26] Sun, Wei, ”Rank-Based Methods for Single-Index Varying Coefficient Models,” Ph.D. dissertation, Department of Statistics, Auburn University, 2017

[27] Sun, Wei, Bindele, Huybrechts F. , Abebe, Ash, and Correia, Hannah, ”General local rank estimation for single-index varying coefficient models.” Journal of Statistical Planning and Inference 202, 57-79, 2019

[28] White, H., “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity,” Econometrica, 48, 817-38. 1980.

[29] Xia, Yingcun & Li, W., ”On Single-Index Coefficient Regression Models,” Journal of the American Statistical Association, 94(448), 1275-1285, 1999

[30] Xue, Liugen & Pang, Zhen, ”Statistical inference for a single-index varying-coefficient model,” Statistics and Computing, 23(5), 589-599, 2013

[31] Yao, Weixin, Lindsay, Bruce G., & Li, Runze, ”Local modal regression,” Journal of Nonparametric Statistics, 24(3), 647-663, 2012

[32] Zhan, Xiaojiang & Hettmansperger, Thomas P., “Bayesian R-estimates in Linear Models,” Statistics and its Interface, Volumue 2, 247-254, 2009.

[33] Zeng, Peng, He, Tianhong & Zhu, Yu, “A Lasso-Type Approach for Estimation and Variable Selection in Single Index Models,” Journal of Computational and Graphical Statistics, Vol.21, Issue 1, 2012.

Appendix A

R Codes

In this appendix we provide two example R codes. The first R code is one of the cases in simulation 3.4.1 which is the linear model with measurement error. The second R code is the main part of simulation 4.4.1 which is the SIM.

```
#R code 1: Simulation 3.4.1#

library(rjags)
library(jagsUI)

#Sample size is 30#
N1 <- 30
N2 <- N1*(N1-1)/2

#True Beta is 1#
beta <- 1

#Conduct 50 repetitions#
N.sim <- 50
nonbayesian.mean.results <- numeric(0)
nonbayesian.median.results <- numeric(0)

nonbayesian.mean.results2 <- numeric(0)
nonbayesian.median.results2 <- numeric(0)

bayesian.mean.results <- numeric(0)
bayesian.median.results <- numeric(0)
naive.results <- numeric(0)

for(k in 1:N.sim)
{

  #Simulation data#
  u <- rnorm(N1, 0, 1)
  modelerror <- rt(N1, 3)
  observed.sig.ux <- 0.1
  x <- runif(N1,-1.5, 1.5)
  y.obs <- beta*x + modelerror
  x.obs <- x + u

  #Function used to pair up the data#
  pairup = function(x,type = "less") {
    x = as.matrix(x)
    n = dim(x)[1]
    a = rep(1:n, rep(n, n))
    b = rep(1:n, n)
    c1 = apply(x, 2,
               function(y){rep(y, rep(length(y), length(y)))})
    c2 = apply(x, 2, function(y){rep(y, length(y))})
    ans = cbind(c1, c2)
    ans = switch(type, less = ans[(a < b), ],
                 leq = ans[a <= b, ], neq = ans)
    ans
  }

  x.non <- pairup(x.obs)
  x.non <- x.non[,2]
  y.non <- pairup(y.obs)
  y.non <- y.non[,2]

  w.obs <- as.matrix(x.obs)
  w.non <- as.matrix(x.non)

  #Use JAGS to get BRank estimate#
  data.jags <- list(N2 = N2, Nrep = 1, Y = y.non, W = w.non,
                    tauu = (1/(observed.sig.ux^2)))
  params <- c("beta", "sigmaeps", "sigmax")


#Model_string.txt#

      for(i in 1:N2){
      Y[i] ~ ddexp(meanY[i], taueps)
      meanY[i] <- beta*X[i]

      W[i] ~ ddexp(X[i], tauu)

      X[i] ~ dnorm(0, taux)
      }
      taueps ~ dgen.gamma(0.5, 0.5, 1)
      taux ~ dgamma(0.5, 2)
      beta ~ dnorm(0, .000001)
      sigmaeps <- 1/sqrt(taueps)
      sigmax <- 1/sqrt(taux)

#End of model_string.txt#

modfile <- "model_string.txt"
jags.pre <- jags(model.file = modfile, data = data.jags,
                 parameters.to.save = params, n.adapt = 100000,
                 n.chains = 3, n.burnin = 50000,
                 n.iter = 200000, n.thin = 50, parallel = TRUE)

nonbayesian.mean.results[k] <- jags.pre$mean$beta
nonbayesian.median.results[k] <- jags.pre$q50$beta

#Use JAGS to get Bayesian estimate#
data.jags2 <- list(N1 = N1, Nrep = 1, Y = y.obs, W = w.obs,
                   tauu = (1/(observed.sig.ux^2)))
params2 <- c("beta", "sigmaeps", "sigmax")

#Model_string2.txt#
    for(i in 1:N1){
    Y[i] ~ dnorm(meanY[i], taueps)
    meanY[i] <- beta*X[i]

    W[i] ~ dnorm(X[i], tauu)

    X[i] ~ dnorm(0, taux)
    }
    taueps ~ dgamma(0.5, 2)
    taux ~ dgamma(0.5, 2)
    beta ~ dnorm(0, .000001)
    sigmaeps <- 1/sqrt(taueps)
    sigmax <- 1/sqrt(taux)
#End of model_string2.txt#

modfile2 <- "model_string2.txt"
jags.pre2 <- jags(model.file = modfile2, data = data.jags2,
                  parameters.to.save = params2, n.adapt = 30000,
                  n.chains = 3, n.burnin = 5000, n.iter = 60000,
                  n.thin = 10, parallel = TRUE)

bayesian.mean.results[k] <- jags.pre2$mean$beta
bayesian.median.results[k] <- jags.pre2$q50$beta

#MCM Estimate#
fit.naive <- lm(y.obs ~ x.obs,x = TRUE)
fit.naive$coef[2]
naive.results[k] <-
  fit.naive$coef[2]*(var(x.obs)/(var(x.obs) - var(u)))

#SIMEX Estimate#
fit.simexQ <- simex(fit.naive,SIMEXvariable = c("x.obs"),
                    observed.sig.ux, fitting.method = "quadratic")
simex.resultsQ[k] <- fit.simexQ$coef[2]
#fit.simexNL <- refit(fit.simexQ, "nonl")
#simex.resultsNL[k] <- fit.simexNL$coef[2]
}

#End of R code 1#

#R code 2: Simulation 4.4.1#

library(rjags)
library(jagsUI)
library(parallel)
library(foreach)
library(doParallel)

N1 <- 200
p <- 4
numKnots <- 6
nsim <- 50
set.seed(7)

```

```
#Create a cluster with # cores available#
cl <- makeCluster(detectCores())

#Register the cluster#
registerDoParallel(cl)

res = foreach(h = 1:nsim,
              .combine = "rbind",
              .packages = c("rjags","jagsUI")) %dopar% {

                x <- matrix(runif(N1*p, -2, 2), N1, p)
                theta0 <- c(2,2,4,1)/5
                z <- x%*%theta0
                y.obs <- 1.5 * sin(pi*z) + rnorm(N1)

                YY <- matrix(0, N1, N1)
                for (i in 1:(N1 - 1)){
                  for (j in (i + 1):N1){
                    YY[j,i] <- y.obs[j,1] - y.obs[i,1]
                  }
                }

                data.jags <- list(N1 = N1, numKnots = numKnots,
                                  Y = y.obs[,1], X1 = x[,1],
                                  X2 = x[,2], X3 = x[,3],
                                  X4 = x[,4])
                params <- c("phi1","phi2", "phi3", "f", "beta0")

                #Model_string1.txt#
                    for (i in 1:N1)
                    {
                    s[i] <- cos(phi1)*X1[i]
                    + (sin(phi1)*cos(phi2))*X2[i]
                    + (sin(phi1)*sin(phi2)*cos(phi3))*X3[i]
                    + (sin(phi1)*sin(phi2)*sin(phi3))*X4[i]

                    f[i] <- beta0 + betas*s[i]
                    + inprod(u[], q[i,])

                    Y[i] ~ dnorm(f[i], taueps)
                    }

                    for (k in 1:numKnots){
                    knot[k] <- ((numKnots+1-k) * min(s[])
                                + k * max(s[]))/(numKnots+1)

                    u[k] ~ dnorm(0,tauU)
                    for (i in 1:N1){
                    q[i,k] <- (s[i] - knot[k])
                    * step(s[i] - knot[k])

                    }
                    }

                    phiMin <- 0
                    phiMax <- 3.141593/2

                    phi1 ~ dunif(phiMin, phiMax)
```

```
    phi2 ~ dunif(phiMin, phiMax)
    phi3 ~ dunif(phiMin, phiMax)

    betas ~ dnorm(0, 1.0E-8)
    beta0 ~ dnorm(0, 1.0E-8)

    tauU ~ dt(0, 1/(25^2), 1)I(0,)
    taueps ~ dgamma(0.5, 2)

    #End of model_string1.txt#

modfile <- "model_string1.txt"
jags.pre1 <- jags(model.file = modfile,
                  data = data.jags,
                  parameters.to.save = params,
                  n.adapt=5000, n.chains=3,
                  n.burnin = 5000,
                  n.iter = 20000,
                  n.thin = 50,
                  parallel = TRUE)


data.jags <- list(N1 = N1, numKnots = numKnots,
                  Y = y.obs[,1],
                  X1 = x[,1], X2 = x[,2],
                  X3 = x[,3], X4 = x[,4])
params <- c("phi1","phi2", "phi3", "f","beta0")

#Model_string2.txt#
    for (i in 1:N1)
    {
    s[i] <- cos(phi1)*X1[i]
    + (sin(phi1)*cos(phi2))*X2[i]
    + (sin(phi1)*sin(phi2)*cos(phi3))*X3[i]
    + (sin(phi1)*sin(phi2)*sin(phi3))*X4[i]

    f[i] <- beta0 + betas*s[i]
    + inprod(u[],q[i,])

    Y[i] ~ ddexp(f[i], taueps)
    }


    for (k in 1:numKnots){
    knot[k] <- ((numKnots+1-k)*min(s[]) +
                  k*max(s[]))/(numKnots+1)

    u[k] ~ dnorm(0,tauU)
    for (i in 1:N1){
    q[i,k] <- (s[i] - knot[k])
    * step(s[i] - knot[k])

    }
    }

    phiMin <- 0
    phiMax <- 3.141593/2

    phi1 ~ dunif(phiMin, phiMax)
    phi2 ~ dunif(phiMin, phiMax)
    phi3 ~ dunif(phiMin, phiMax)

    betas ~ dnorm(0, 1.0E-8)
    beta0 ~ dnorm(0, 1.0E-8)

    tauU ~ dt(0, 1/(25^2),1)I(0,)
    taueps ~ dgen.gamma(0.5, 0.5, 1)

    #End of model_string2.txt#

modfile <- "model_string2.txt"
jags.pre2 <- jags(model.file = modfile,
                  data = data.jags,
                  parameters.to.save = params,
                  n.adapt = 5000, n.chains=3,
                  n.burnin = 5000,
                  n.iter = 20000,
                  n.thin = 50, parallel = TRUE)

data.jags <- list(N1 = N1, numKnots = numKnots,
                  Y = y.obs[,1], YY=YY, X1=x[,1],
                  X2=x[,2], X3=x[,3], X4=x[,4])

params <- c("phi1","phi2", "phi3", "f","beta0")

#model_string3.txt3
    for (i in 1:N1)
    {
    s[i] <- cos(phi1)*X1[i]
    + (sin(phi1)*cos(phi2))*X2[i]
    + (sin(phi1)*sin(phi2)*cos(phi3))*X3[i]
    + (sin(phi1)*sin(phi2)*sin(phi3))*X4[i]

    f[i] <- betas*s[i] + inprod(u[], q[i,])
    beta0[i] <- Y[i] - f[i]
    }

    for (i in 1:(N1 - 1)){
    for (j in (i + 1):N1){
    ff[j, i] <- f[j] - f[i]
    YY[j, i] ~ ddexp(ff[j, i], taueps)
    }
    }

    for (k in 1:numKnots){
    knot[k] <- ((numKnots+1-k)*min(s[])
                + k*max(s[]))/(numKnots+1)

    u[k] ~ dnorm(0, tauU)
    for (i in 1:N1){
    q[i, k] <- (s[i] - knot[k])
    * step(s[i] - knot[k])

    }
    }

```

```
                    phiMin <- 0
                    phiMax <- 3.141593/2

                    phi1 ~ dunif(phiMin, phiMax)
                    phi2 ~ dunif(phiMin, phiMax)
                    phi3 ~ dunif(phiMin, phiMax)

                    betas ~ dnorm(0, 1.0E-8)

                    tauU ~ dt(0, 1/(25^2), 1)I(0,)
                    taueps ~ dgen.gamma(0.5, 0.5, 1)
                    #End of model_string3.txt#

                modfile <- "model_string3.txt"
                jags.pre3 <- jags(model.file = modfile,
                                  data = data.jags,
                                  parameters.to.save = params,
                                  n.adapt=5000, n.chains=3,
                                  n.burnin = 5000, n.iter = 20000,
                                  n.thin = 50, parallel = TRUE)
                }

stopCluster(cl) # shut down the cluster
folds <- 1:nsim
}
```