ABSTRACT

LIN, CHEN-YEN. Some Recent Developments in Parametric and Nonparametric Regression Models. (Under the direction of Hao Helen Zhang and Howard D. Bondell.)

We present several developments in variable selection techniques for parametric and nonparametric regression models in this dissertation. We begin the series of discussions from a traditional linear model. In recent years, analysis of high-dimensional data has become a routine in modern statistics. High-dimensional data brings about new opportunities and also new challenges for many classical procedures. In Chapter 1, we review a classic yet popular variable selection method, forward selection. We propose a perturbed forward selection to alleviate several difficulties that forward selection experiences when predictors are high-dimensional. As another manifestation of the variable selection ensemble approach, the proposed method requires running forward selection on multiple perturbed datasets and aggregating the results to provide a powerful and stable selection path. In Chapters 2 and 3, we shift attention to nonparametric models. In nonparametric regression, variable selection for multivariate regression is a challenging task. The success of penalized regression, especially the COSSO penalty, opens a door for joint estimation and selection. We first study variable selection for nonparametric quantile regression. We present a novel computational algorithm to implement the adaptive COSSO penalty in quantile regression. Moreover, for better parameter tuning, we introduce a bootstrap-type degrees-of-freedom estimate. The promising numerical results serve as another testament of the success of COSSO penalty. To better promote the applicability and to establish the asymptotic properties of COSSO for various regression models, we propose a nonparametric least squares approximation procedure in Chapter 3 that provides a unified framework to do variable selection and function estimation. In addition, the proposed procedure also enjoys lower computational cost, making it a desirable alternative for existing COSSO-type methods. We observe very encouraging numerical results and our future work is to study the asymptotic properties of the proposed least squares approximation procedure.

Some Recent Developments in Parametric and Nonparametric Regression Models

by
Chen-Yen Lin

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina

2012

APPROVED BY:

Leonard A. Stefanski	Yichao Wu
Hao Helen Zhang Co-chair of Advisory Committee	Howard D. Bondell Co-chair of Advisory Committee

BIOGRAPHY

Lin, Chen-Yen was born in Chiayi, Taiwan and grew up in southern Taiwan. He earned his Bachelor degree in Economics in 2003 from Fu-Jen Catholic University and Master degree in Statistics in 2007 from National Chengchi University. After his master study, he worked for Institute of Statistical Science, Academia Sinica as a research assistant for one year. He went to North Carolina State University to pursue his doctoral degree in Statistic in 2008. After graduation, he will continue his research as a postdoctoral fellow at the Department of Biostatistics and Bioinformatics at Duke University School of Medicine.

ACKNOWLEDGEMENTS

First of all, I want to take this opportunity to express my great appreciation to my advisors Dr. Hao Helen Zhang and Dr. Howard D. Bondell. I benefit so much from their profound insight into statistics and their professionalism. Helen and Howard have been very generous and patient to me. I feel extremely fortunate to have them as my thesis advisors.

My sincere thanks also go to Dr. Leonard A. Stefanski and Dr. Yichao Wu for joining my committee and providing many suggestions on my research. I wish to thank other faculty members in the statistics department for their excellent lectures.

Finally, special thanks to all the friends I love and friends who love me from Taiwanese Student Association. The journey will be so much less colorful without you.

"Faithful friends are a sturdy shelter: whoever finds one has found a treasure." (Sirach *6:14*)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Refining Forward Selection in High-Dimensional Feature Space by Perturbation

1.1 Introduction

We consider a high-dimensional linear regression model

$$y_i = \beta_1 x_i^{(1)} + \ldots + \beta_p x_i^{(p)} + \varepsilon_i,\ i = 1, \ldots, n, \tag{1.1}$$

where y_i is the response, $\boldsymbol{x}_i = (x_i^{(1)}, \ldots, x_i^{(p)})$ is a p-dimensional vector of predictors, $p \gg n$, and ε_i is the random error with mean zero and finite variance. We assume that the response and each predictor are centered so there is no intercept in (1.1).

The motivation of this research arises from current challenges in high-dimensional regression. For instance, one scientific goal of a microarray experiment is to identify a set of genes that are related to a continuous phenotypic response. The difficulties and challenges from this type of study are two-fold: limited sample size; and complex correlation structure among predictors. In a typical microarray experiment, the number of arrays is usually on the order of tens while the number of predictors is tens of thousands. In addition, in such a high-dimensional model, not only can spurious correlations lead to incorrect scientific findings (Fan and Lv, 2010) but the correlations between genes within the same biological pathway complicates the selection process.

The majority of variable selection methods revolve about the notion of selection consistency and examine how often a method identifies the correct model. However, in the modern world of high-dimensional data, a scientist does not expect to identify the correct predictors with no mistakes. In many instances, a scientist would simply prefer a properly ranked list of candidate predictors and hope that the important ones would tend to be ranked at or near the top of the list. Our goal is to obtain a ranked list whose ordering improves upon the ordering obtained by the existing methods. As also remarked by Xin and Zhu (2012), the task of ranking is the most fundamental. Once the variables are ranked, from a decision-theoretic framework, the choice of thresholding has more to do with one's belief on the tradeoff between false positives and false negatives.

For any selection procedure that generates a ranked list or a sequence of candidate models, such as forward selection or penalized regression, despite many information criteria having been proposed, it remains a highly debated topic how to pick a final model from the selection path. Even in the traditional large n small p situations, the correct model may not be an element in the selection path (Leng et al., 2006; Wang, 2009), making selection consistency less realistic in the high-dimensional case. More discussion on the selection consistency of penalized regression methods can be found in Zhao and Yu (2006) and Fan and Lv (2010) and references therein. When selection consistency is not feasible, a pertinent alternative is to study if a selection procedure possesses the sure screening property, i.e. all important predictors would be included with probability going to one (Fan and Lv, 2008).

In this study, we revisit a classic yet popular selection procedure, forward selection (FS). Recently, Wang (2009) studied the sure screening property of FS and showed theoretically and numerically that FS can consistently detect all important predictors even if the number of predictors is substantially larger than the sample size. Despite that FS enjoys such desirable property, the method has several limitations. For instance, resulting from its greedy search algorithm, FS tends to eliminate other informative predictors if they are correlated with the ones that are in the current model (Efron et al., 2004). In a high-dimensional setup, Donoho and Stodden (2006) showed that there exists a breakdown point for standard model selection procedures including FS and LASSO (Tibshirani, 1996, 2011) when the number of variables exceeds the sample size. Moreover, both FS and LASSO can only identify at most n predictors before it saturates when $n \ll p$.

We address these limitations by a re-weighting approach. Motivated from the mini-

mand perturbation of Jin et al. (2001), we propose a computationally-intensive method, which we call the perturbed FS. The notion of minimand perturbation is originally introduced to derive the sampling distribution of some parameter estimates in parametric models. In this work, we explore the perturbation technique in order to enhance the stability in variable selection. Our perturbation method is reminiscent of a weighted least squares (WLS), as it can be viewed as randomly weighting the observations. With random weights, the WLS method bears some similarity with Bayesian bootstrap (Rubin, 1981) and Bayesian Bagging (Clyde and Lee, 2001).

Compared to the original FS, the main advantage of the proposed method is that: the new method no longer depends on a greedy search hence can better handle correlated predictors and identify more predictors than sample size. More importantly, as we demonstrate later in the article, the perturbed FS provides a competitive, and often superior, variable ordering and prediction accuracy. Obviously the price we pay is increased computational intensity. However, as will be explained later, the procedure involves applying FS to multiple perturbed data and the implementation of the procedure does not require communication between different tasks and therefore can be facilitated by taking advantage of parallel computing (Knaus et al., 2009).

Our proposed method is based on repeatedly applying FS on multiple perturbed data and produces an aggregated importance indicator for each predictor. This philosophy has a close proximity to a higher-level notion of variable-selection ensemble (VSE) (Xin and Zhu, 2012). Ensemble methods were originally proposed in the machine learning literature, such as bagging (Breiman, 1996) and random forests (Breiman, 2001). More recently, ensemble methods have become popularized in the variable selection context, for example, random LASSO (Wang et al., 2011) and stability selection (Meinshausen and Bühlman, 2010). Both methods conceptually generate many bootstrap samples and apply LASSO algorithm repeatedly to produce a more stable and powerful procedure. Thus, Random LASSO, stability selection and our perturbed FS can all be viewed as different manifestations of VSE.

The remainder of the article is organized as follows. Section 2 reviews the classical FS algorithm and introduces our perturbed FS method. We illustrate the performance of our method using simulation and a real data in Section 3 and provide a summary of our findings in Section 4.

1.2 Forward Selection

Let $\{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$, be the observed random sample from a population where the relationship between y_i and $\boldsymbol{x}_i$ can be described through a linear function in (1.1). Denote $\mathcal{M} = \{j_1, \ldots, j_{p^*}\}$ as the model containing $x_i^{(j_1)}, \ldots, x_i^{(j_{p^*})}$ as relevant predictors and $|\mathcal{M}|$ as the cardinality of the set. Further denote the true model as $\mathcal{M}_T = \{j : \beta_j \neq 0\}$ where we assume $|\mathcal{M}_T| = p_0 \ll p$.

1.2.1 Forward Selection Path

The original forward selection algorithm can be summarized in the following steps

Step 0: (Initialization) Set $\mathcal{S}^{(0)} = \{\emptyset\}$.

Step 1: In the k-th step ($k \geq 1$), for all $j \in \{1, \ldots, p\} \backslash \mathcal{S}^{(k-1)}$, consider a candidate model $\mathcal{S}^{(k-1)} \cup \{j\}$ and compute its sum of squared error $\mathrm{SSE}_j^{(k-1)}$. Identify which predictor results in the smallest sum of squared error, say $j_*^k = \underset{j \in \{1,\ldots,p\} \backslash \mathcal{S}^{(k-1)}}{\arg\min} \mathrm{SSE}_j^{(k-1)}$. Then update the model at the k-th step $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)} \cup \{j_*^k\}$.

Step 2: Increase loop index k by 1 and go back to Step 1 until $k = n$.

In a high-dimensional setup, the FS algorithm experiences several difficulties. For instance, after repeating Step 1 n times, the fitted model will have sum of squared error zero and thus the procedure stops with at most n predictors in the final fitted model. Besides, in a high-dimensional feature space, the spurious correlation can easily mask the true signals and mislead the FS to include a noise variable. Furthermore, FS fails to provide a stable selection result when the degree of collinearity is high. The orders of the correlated predictors entering the selection path are sensitive to the data on hand. To enhance the stability of FS in an ideal situation, we apply FS on multiple datasets generated from the same population and observe the commonality among their corresponding selection paths. However, there is only one available data in practice. Partitioning the data into several small portions is not an efficient way to use the data and usually the result derived from each portion would be even more unreliable. Thus, we initiate our new method from data perturbation.

1.2.2 Perturbed Forward Selection Path

To generate multiple perturbed datasets, one popular method is the bootstrap. However, bootstrap-type methods suffer from some immediate difficulties when $n < p$. For instance, bootstrapping residuals is not practical since all residuals are zeros. Moreover, nonparametric bootstrap will produce less than n unique observations, making the number of identifiable predictors less than n, and can in fact be much less. More recently, Meinshausen and Bühlman (2010) proposed a stability selection procedure which randomly chooses a subsample of size $n/2$ as a method to stabilize the LASSO penalization. This method has the limitation that the number of recoverable predictors becomes $n/2$.

Due to the limitations of bootstrap and related data perturbation methods, we consider an alternative by perturbing the objective function as in Jin et al. (2001). In the least squares context, the objective function we aim to minimize is given by

$$\min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \tag{1.2}$$

where $\boldsymbol{W} = \text{diag}(w_1, \ldots, w_n)$ is a diagonal matrix containing weights for each observation. The original unperturbed data would have $\boldsymbol{W} = \boldsymbol{I}$. We adopt a random weight generated from an underlying distribution $F(w)$, which can be viewed as a perturbation to the objective function. In principle, any non-negative random variable can be used as weight and, as remarked in Jin et al. (2001), the solution is robust to the choice of distribution function. Different choices of $F(\cdot)$ allows us to make several interesting analogies to existing methods. For instance, when $F(\cdot)$ is the Bernoulli distribution function with success probability 0.5, i.e. we expect to retain $n/2$ observations, the perturbation method is closely related to aforementioned stability selection. We propose, instead, to consider a continuous weight, more specifically, the exponential weight. When $F(\cdot)$ is the distribution function of an exponential random variable, it is essentially the same as assigning a random Dirichlet weight to each observation, giving some resemblance to Bayesian Bootstrap (Rubin, 1981) and Bayesian bagging (Clyde and Lee, 2001).

Although we initiate the new method from perturbing the objective function, the WLS problem in (1.2) is equivalent to an OLS problem by multiplying each observation by its corresponding square root of the weight. Thus, as an equivalent formulation, we consider a multiplicative perturbation method to generate B datasets. Let $w_i^{(b)} \overset{iid}{\sim} F(w) = 1 - \exp(-w),\ i = 1, \ldots, n;\ b = 1, \ldots, B$, and denote the b-th perturbed

dataset by $(y_i^{(b)}, \boldsymbol{x}_i^{(b)}) = \sqrt{w_i^{(b)}}(y_i, \boldsymbol{x}_i)$. After generating B perturbed datasets, we apply the aforementioned FS algorithm on each of them and store its saturated model $\mathcal{S}^{(n)}(b)$. Denote $\hat{\pi}_j = B^{-1} \sum_{b=1}^{B} I(j \in \mathcal{S}^{(n)}(b)),\ j = 1, \ldots, p$, as the empirical probability of selecting the j-th predictor among B perturbations, then the perturbed FS path is given by ranking the empirical selection probabilities in a descending order.

The computation cost depends on the number of perturbed datasets and a naive programming algorithm is sequentially applying FS on each of them. In light of the fact that this procedure does not require communication between the FS computations but performs each task separately, a sophisticated yet efficient programming technique is to take advantage of parallel computing. Most commercially available computers nowadays are equipped with two to eight processing cores. To fully exploit the devices, an efficient algorithm should reduce to multiple parallel tasks, each accessing a specific dataset. We implemented a parallel computing algorithm to accelerate the procedure in `R` using the `snowfall` package (Knaus et al., 2009). The supporting `R` code is available from the authors upon request.

To visually show the effectiveness of data perturbation, we use a microarray experiment of Scheetz et al. (2006) as a motivating example. This gene expression dataset consists of 120 arrays, each array contains 31,042 probe sets (Affymetric GeneChip Rat Genome 230 2.0 Array). The complete gene expression data is available at Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo; accession number GSE5680). The primary objective of this study is to identify which gene expressions are related to that of gene TRIM 32, which is recently found to cause Bardet-Biedl syndrome (Chiang et al., 2006).

The probe ID associated with the response, TRIM32, is 1389163_at. To identify which genes are correlated with TRIM32, we regress the expression of TRIM32 on the remaining probes. Since this is real data, to demonstrate how our method copes with noise variables, we first choose 4 probes whose magnitude of marginal correlations with the response are the largest. Then we randomly select another 1,992 probes from the remaining 31,033 and randomly permute their values across the arrays. The purpose of random permutation is to create a scenario where the 8 unpermuted probes are treated as true signals whereas those permuted probes are noise. In Figure 1.1, the empirical selection probabilities $\hat{\pi}_j,\ j = 1, \ldots, 2000$, are plotted on the y-axis as a function of the number of perturbations on the x-axis. Without any perturbation, the original FS can only identify

3 unpermuted genes out of 8. However, after a reasonable number of perturbations, most of the unpermuted genes stand out and the separation between permuted and unpermuted genes becomes clear toward the end. This encouraging finding justifies the use of perturbation method to achieve better variable ordering in a high-dimensional feature space.

We give a formal description of the perturbed FS and the computational algorithm here. The perturbed FS algorithm can be summarized in the following steps.

Step 0: Initialization. Set $\mathcal{S}_P^{(0)} = \{\emptyset\}$.

Step 1: In the k-th step ($k \geq 1$), identify the predictor with the largest empirical selection probability, say $j_k^* = \underset{\{1,\ldots,p\}\setminus\mathcal{S}_p^{(k-1)}}{\arg\max}\ \hat{\pi}_j$. Then update the model at the k-th step $\mathcal{S}_p^{(k)} = \mathcal{S}_p^{(k-1)} \cup \{j_k^*\}$.

Step 2: Increase loop index k by 1 and go back to Step 1 until all predictors are included in the path, i.e. $k = p$.

Considering the finite number of perturbations may not guarantee an unique maximum in Step 1, there are two types of situations that needs special treatments, including

Case 1: ($\hat{\pi}_j = \hat{\pi}_k > 0$, for some j, k): We break the tie by the average step they enter the FS. For a given perturbed path that selects the j-th predictor, we not only know $j \in \mathcal{S}^{(n)}(b)$ but the step it is included. Thus, we let the j-th predictor enter the perturbed path first if, in average, it takes fewer steps to enter each perturbed path.

Case 2: (the set $\{j : \hat{\pi}_j = 0\}$ is not empty): We break the tie by the magnitude of their marginal correlations with the response. The larger the absolute correlation, the earlier it enters the perturbed FS path.

Intuitively, when a sufficiently large number of perturbations is used, there should not be ties between predictors and each predictor, even if it is a noise one, will be selected at least once. Following the logic, another way to construct a perturbed FS path is continuing perturbing the data until each predictor has its unique positive empirical selection probability. This requirement, however, is computationally infeasible when p is

in the order of tens of thousands. For a reasonable number of perturbations, we adopt the tie-breaking technique as introduced before to save computation cost.

By construction, of several major differences between the original FS and the perturbed FS, one of which is the traditional FS path can only rank the most important n predictors; whereas the perturbed path provides a comprehensive rank for all p predictors. For fair comparison, for those predictors which are not selected by the original FS, we rank them according to their marginal correlations and then append the ordered path into the original FS path so that both of the original FS and perturbed paths have length p.

1.3 Numerical Study

1.3.1 Simulation Models

In this section, we demonstrate the perturbed FS and compare it to the original FS using simulations. Five different examples are considered in this study.

Example 1 (Independent Features): We start from an example that is similar to the one used in Fan and Lv (2008). There are $p = 1000$ predictors and $p_0 = 10$ non-zero coefficients. Each predictor is independently generated from standard normal distribution. The first p_0 coefficients are non-zero and are given by $\beta_j = (-1)^{U_j}(4 \log n/\sqrt{n} + |Z_j|),\ j = 1, \ldots, p_0$, where U_j follows a Bernoulli distribution with success probability 0.4 and Z_j is another independent random variable following a standard normal distribution.

Example 2 (Autoregressive): Following a similar setting as that in Example 1, we let $p = 1000$ and $p_0 = 8$ but the correlation between predictors having an autoregressive structure with pairwise correlation $\text{cor}(x_i^{(j)}, x_i^{(k)}) = 0.7^{|j-k|},\ \forall j \neq k$. Similarly, the first p_0 coefficients are non-zero and generated in the same fashion as before.

Example 3 (Compound Symmetry): To further examine the performance of the perturbed FS path, we consider a higher dimensional example. The number of predictors becomes $p = 5000$ and only the first 8 coefficients are non-zero with constant value of 5. We consider another common correlation structure, compound symmetry, so the pairwise correlation becomes $\text{cor}(x_{ij}, x_{ik}) = 0.5,\ \forall j \neq k$.

Example 4 (Factor Model): This example is based on Meinshausen and Bühlman (2010) with $p = 1000$ and $p_0 = 4$. Let ϕ_1, ϕ_2 be two latent factors that independently come from $\mathcal{N}(0, 1)$. Then each predictor x_{ij} is generated as $x_{ij} = f_{ij,1}\phi_{i1} + f_{ij,2}\phi_{i2} + \eta_{ij}$, where $f_{ij,1}$, $f_{ij,2}$ and η_{ij} have *i.i.d.* standard normal distributions for all $j = 1, \ldots, p$. The four locations of non-zero coefficients are randomly chosen and the coefficients are generated from uniform $(0, 1)$.

Example 5 (Diverging parameters): In the previous examples, the true model sizes are fixed. We consider a different situation where p_0 diverges with the sample size (Zou and Zhang, 2009). More specifically, we adopt the similar setup as Example 1 but let $p = 5000$ and $p_0 = \lfloor \sqrt{n} \rfloor$.

In Examples 1-4, we consider two sample sizes and two theoretical $R^2 = \frac{\mathrm{Var}(\boldsymbol{x}_i\boldsymbol{\beta})}{\mathrm{Var}(y_i)}$ combinations. As for Example 5, we fix $R^2 = 0.6$ and vary the sample size. Later we use the notation (n, p, p_0, R^2) to denote the combination of sample size, number of predictors, number of non-zero coefficients and theoretical R^2. Regarding the number of perturbations, the exploratory experiment shown in Figure 1.1 suggests the selected probability stabilizes moderately fast. So we use $B = 200$ in the simulation. We also tried $B = 300$, but the results were comparable. We run each simulation scenario 100 times and report the summary statistics and their associated standard error.

To evaluate the quality of variable ordering, considering the candidate model at each step of the selection path, we compute the true positives, the number of informative predictors included in the current step, and the false positives, the number of noise predictors included in the current step. The receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate, or equivalently the sensitivity against one minus specificity, on a two-dimensional plane, is a common tool to illustrate the relationship between type I error and power.

Like the ROC curve delineates the trade-off between sensitivity and specificity, the precision recall (PR) curve, which plots one minus False Discovery Rate (FDR) against true positive rate on a two-dimensional plane, provides another perspective to examine the relationship between FDR and power. In high-dimensional inference problem, one is usually more concerned about FDR rather than type I error. Thus the PR curve could be a more sensible assessment in our study. To give a point summary of the curve, we use the area under curve as a measurements of the overall performance of the selection path.

1.3.2 Simulation Result

Based on the simulation result summarized in Table 1.1, the perturbed FS evidently outperforms the original FS across all scenarios. In Examples 1 and 5, the area under PR curve suggests as large as 30% improvement can be achieved by using the perturbed FS. In Examples 2-4, where we consider correlated predictors, greater improvement can be expected. This observation is coherent with the theoretical property of the original FS. The greedy algorithm hinders other prominent predictors from entering the selection path because of their correlation with the ones that are already included in the current model. This greedy nature can be illustrated in Figure 1.2. In Figure 1.2, we demonstrate how the condition number of the design matrix changes with the model size in the second simulation example. Since the original FS tends to include an additional predictor which is less correlated to those in the current model, it naturally leads to a design matrix with relatively smaller condition number.

Complementary to a point summary given in Table 1.1, Figure 1.3 provides a comprehensive visual comparison between these two competing methods. Shown in Figure 1.3 are averaged ROC curves over 100 simulated data in Example 1 with $n = 150$ and $R^2 = 0.5$. As can be seen from the left panel, the ROC curve of the perturbed FS completely dominates that of the original FS, particularly when specificity is greater than 0.3. When specificity becomes less and less, the separation between these two lines quickly vanishes, which is the reason why their area under ROC curves do not differ by a large margin as that in the area under PR curve. Nonetheless, when specificity is 0.6, it implies the model size is around 400, which is not feasible since sample size is only 150. In practice, our attention is usually drawn toward the beginning of the path or a more practical model size. In the right column of the plot is the same ROC curve but zoom in the region where specificity is greater than 0.9, or model size is less than 100. In the region of interest, the effectiveness of perturbed FS becomes transparent. Therefore, it is clear that most improvement of our new method comes from a better variable ranking in the beginning of the path.

To better illustrate how the perturbed FS improve the variable ranking, we can directly compare the true positives at each step of the path. In Figure 1.4, we draw the boxplots for true positives at model sizes 1 to 15 in Example 3. The perturbed path constantly has larger true positives at any given model size, which is consistent with higher power shown in Figure 1.3. An evident separation between two boxes suggests the

perturbed FS provides superior ranking.

1.3.3 Real Data

To examine the real data application, we analyzed two microarray datasets: the rat array (Scheetz et al., 2006) and the inbred mouse array (Lan et al., 2006). The rat array is described in Section 1.2.2. The inbred mouse data consists of 60 arrays, 31 female and 29 male mice, and each array measures the expression values of 22,690 genes. A continuous phenotypic variables measured by RT-PCR, stearoyl-CoA desaturase 1 (SCD1), is used as the response.

We first screen down the number of genes to 2,000 and 1,999, respectively, using sure independent screening (Fan and Lv, 2008). For the inbred mouse data, we also include gender as an additional predictor so both datasets consist of 2,000 potential predictors.

The performance of our perturbed FS algorithm is compared to the original FS using out-of-sample prediction. To assess prediction accuracy, we first split the data into two folds, training and testing sets, accounting for 80% and 20% of the full data, respectively. We apply the original FS and perturbed FS algorithm on the training set and estimate regression parameters each step of the paths, then apply the estimated model parameters on the testing set and evaluate the mean squared prediction error (MSPE). This process will be carried out 100 times and averaged.

In Figure 1.5, we compute the MSPE from model sizes 1 to 40. In addition, the MSPE from the null model, the model without any predictor, is also provided in the plot as a baseline performance. From Figure 1.5, the MSPE of the perturbed FS is significantly better than that of the original FS in both datasets. The performance is similar in small model sizes, suggesting both methods can identify some strong signals in the beginning. However, the original FS can not continue to identify useful predictors to improve prediction accuracy.

1.4 Discussion

We propose a perturbed FS method to enhance and improve the original FS. The proposed method explores the applicability of minimand perturbation method in the variable selection context. As another testament of the powerfulness of ensemble approach, the novel selection path, which is constructed by empirical selection probabilities, success-

fully alleviates several limitations of the original FS. The number of identifiable variable is no loner limited by the sample size as that in the original FS. Moreover, the ordered variable list has the power to rank important predictors ahead of those irrelevant ones. Simulation studies suggest that the perturbed FS has superior selection path than the original FS, and the real analysis of two microarray datasets shows the sound prediction performance in practice.

In this article, we do not directly address how to select a final model from the path. More recently, EBIC (Chen and Chen, 2008) has become popularized in the high-dimensional selection problem. One issue of EBIC is that it targets at controlling FDR, so that its finite sample performance tends to be conservative (Wang, 2009). The MSPE curve from the real data analysis suggests a valley-shape pattern, implying a out-of-sample or cross-validated type of prediction error could serve as a guidance for determining a "best" model. Nevertheless, as remarked by Xin and Zhu (2012), what really matters for a VSE procedure is the variable ranking. Selecting a final model will require a certain thresholding rule but it all depends on researchers' prior believes. Thus, we are more concerned about the quality of the path throughout the paper.

Table 1.1: Average area under ROC and PR curves in 5 simulation examples over 100 runs. The standard error is given in the parentheses.

		ROC		PR	
	(n, p, p_0, R^2)	Original FS	Perturbed FS	Original FS	Perturbed FS
Example 1	(100,1000,10,0.50)	0.846 (0.006)	0.867 (0.007)	0.263 (0.016)	0.346 (0.015)
	(100,1000,10,0.80)	0.971 (0.004)	0.989 (0.002)	0.792 (0.021)	0.839 (0.012)
	(150,1000,10,0.80)	0.891 (0.005)	0.932 (0.005)	0.540 (0.020)	0.606 (0.014)
	(150,1000,10,0.80)	0.998 (0.001)	1.000 (0.000)	0.984 (0.004)	0.993 (0.002)
Example 2	(75,1000,8,0.50)	0.833 (0.012)	0.854 (0.012)	0.326 (0.017)	0.425 (0.023)
	(75,1000,8,0.80)	0.906 (0.008)	0.947 (0.007)	0.569 (0.022)	0.694 (0.022)
	(100,1000,8,0.80)	0.842 (0.011)	0.896 (0.009)	0.404 (0.017)	0.507 (0.021)
	(100,1000,8,0.80)	0.952 (0.006)	0.978 (0.004)	0.750 (0.019)	0.816 (0.018)
Example 3	(100,5000,8,0.90)	0.884 (0.005)	0.890 (0.008)	0.089 (0.010)	0.300 (0.017)
	(100,5000,8,0.95)	0.915 (0.005)	0.961 (0.004)	0.170 (0.021)	0.521 (0.019)
	(150,5000,8,0.90)	0.928 (0.004)	0.974 (0.003)	0.243 (0.021)	0.598 (0.019)
	(150,5000,8,0.95)	0.983 (0.003)	0.998 (0.001)	0.533 (0.025)	0.926 (0.009)
Example 4	(150,1000,4,0.50)	0.738 (0.014)	0.767 (0.016)	0.269 (0.023)	0.351 (0.027)
	(150,1000,4,0.80)	0.868 (0.012)	0.887 (0.012)	0.613 (0.028)	0.695 (0.025)
	(200,1000,4,0.50)	0.766 (0.015)	0.824 (0.014)	0.383 (0.030)	0.467 (0.028)
	(200,1000,4,0.80)	0.899 (0.011)	0.917 (0.009)	0.725 (0.024)	0.817 (0.017)
Example 5	$(200,5000,\lfloor\sqrt{200}\rfloor,0.60)$	0.932 (0.004)	0.929 (0.005)	0.482 (0.018)	0.545 (0.014)
	$(400,5000,\lfloor\sqrt{400}\rfloor,0.60)$	0.958 (0.003)	0.979 (0.002)	0.778 (0.012)	0.800 (0.010)
	$(800,5000,\lfloor\sqrt{800}\rfloor,0.60)$	0.980 (0.002)	0.994 (0.001)	0.921 (0.006)	0.925 (0.005)

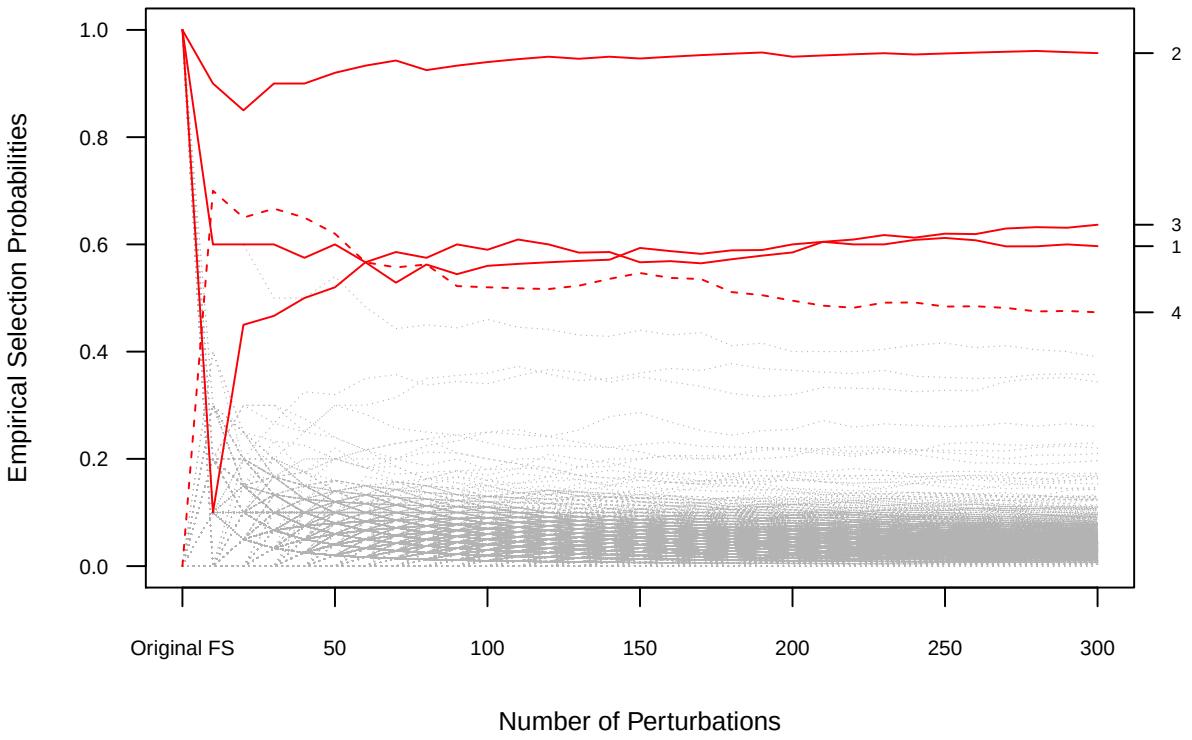

Figure 1.1: The empirical selection probability as a function of number of perturbations. The 8 unpermuted genes are shown in red lines (three solids lines represent the ones that are selected by the original FS); while the permuted genes are shown in grey dot lines.

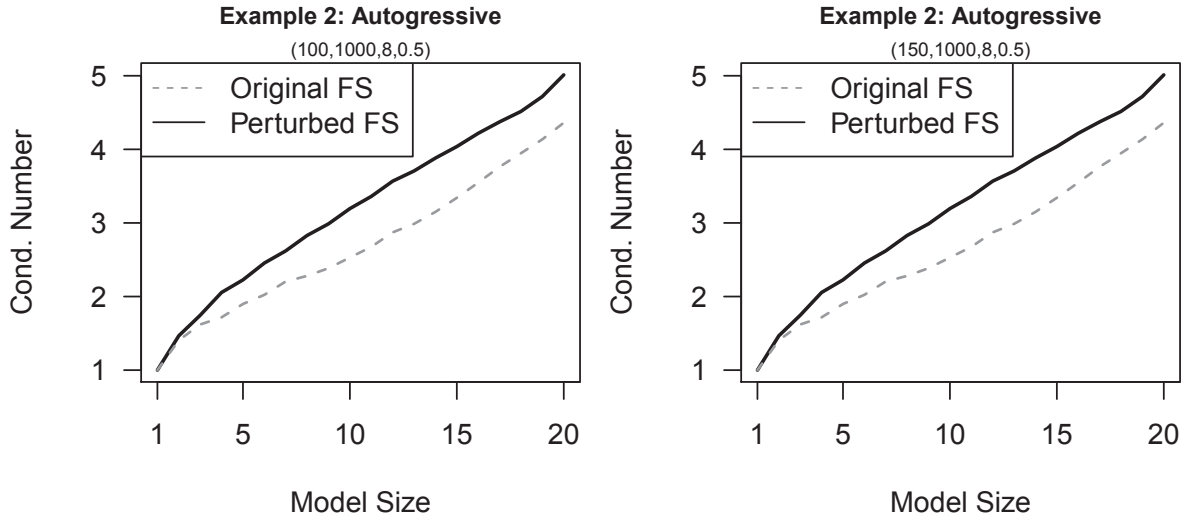

Figure 1.2: Condition number at different model sizes in simulation Example 2. The solid line and broken line represent the perturbed FS and the original FS, respectively.

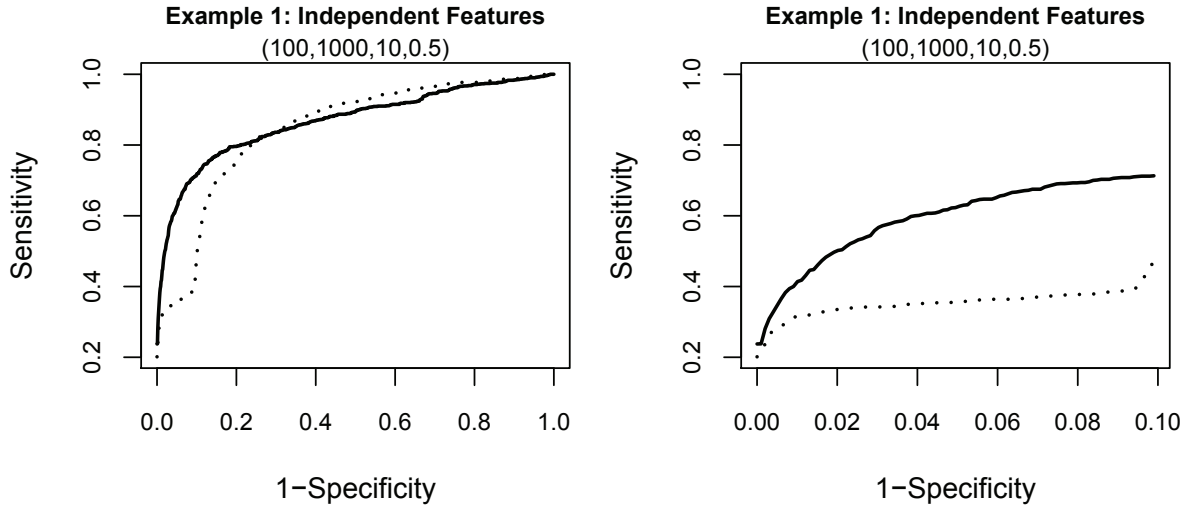

Figure 1.3: Averaged ROC curves for Example 1 with $n = 150$ and $R^2 = 0.5$. The plot on the left panel is the complete ROC curve whereas that on the right is the same curves but zooms in the region where the Specificity is greater than 0.9.

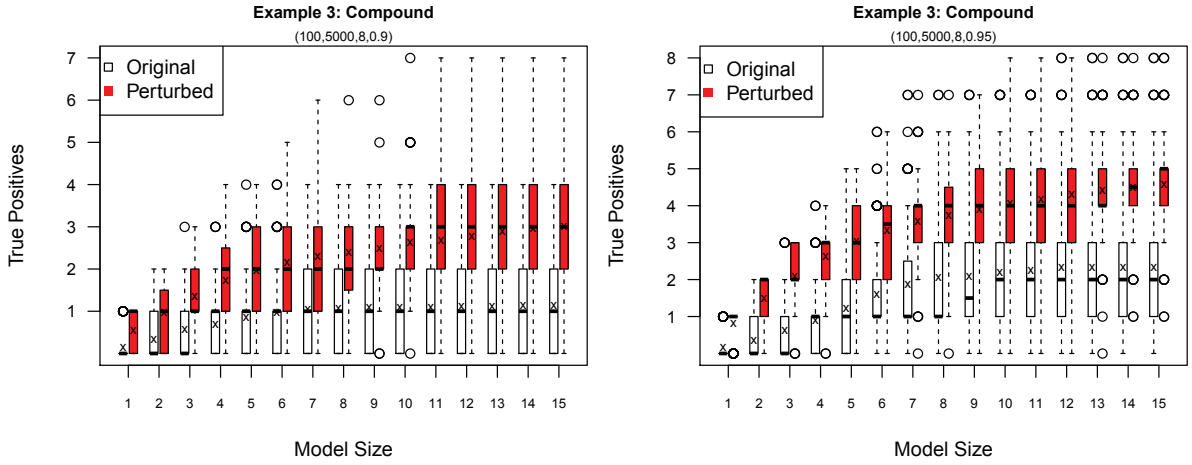

Figure 1.4: True positives at different model sizes in Examples 3. The white box and red box represent the original FS and perturbed FS, respectively. Cross sign represents the average over 100 runs. The left panel is the R^2 =0.90 and the right panel is the R^2 =0.95.

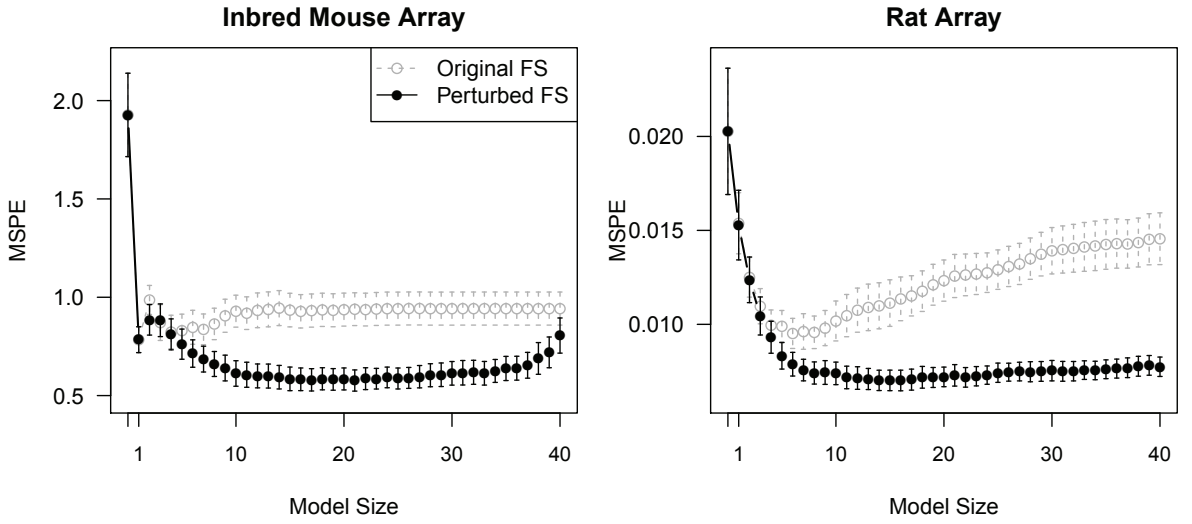

Figure 1.5: Mean squared prediction error curve as a function of model size. The dot points represent the average over 100 random splits and the vertical bars represent the average plus/minus two standard errors.

Chapter 2

Variable Selection for Nonparametric Quantile Regression via Smoothing Spline ANOVA

2.1 Introduction

Quantile regression, as a complement to classical least square regression, provides a more comprehensive framework to study how covariates influence not only the location but the entire conditional distribution (Koenker, 2005). In quantile regression problems, the primary interest is to establish a regression function to reveal how the $100\tau\%$ quantile of the response y depends on a set of covariates $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(p)})$. A parametric form of regression function is often assumed for convenience of interpretation and lower computational cost. While a linear regression function is studied in Koenker and Bassett (1978) and numerous follow-up studies, Procházka (1988) and Jurečková and Procházka (1994) explored nonlinear regression; see Koenker and Hallock (2001) and Koenker (2005) for a comprehensive overview.

As much as the parametric assumption enjoys a simple model structure and lower implementation cost, it is not flexible enough and hence carries the risk of model misspecifications for complex problems. For a single predictor model, Koenker et al. (1994) pioneered nonparametric quantile regression in spline models, in which the quantile func-

tion can be found via solving the minimization problem

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \rho_\tau(y_i - f(x_i)) + \lambda V(f'), \tag{2.1}$$

where $\rho_\tau(\cdot)$ is the so-called "check function" of Koenker and Bassett (1978),

$$\rho_\tau(t) = t[\tau - I(t < 0)],\ \tau \in (0,1), \tag{2.2}$$

λ is a smoothing parameter and $V(f')$ is the total variation of the derivative of f. Koenker et al. (1994) showed that the minimizer is a linear spline with knots at the design points $x_i, i = 1, \ldots, n$, provided that the space $\mathcal{F}$ is an expanded second-order Sobolev space defined as

$$\mathcal{F} = \left\{ f : f(x) = a_0 + a_1 x + \int_0^1 (x-y)_+ d\mu(y),\ V(\mu) < \infty, a_i \in \mathbb{R}, i = 0, 1 \right\}, \tag{2.3}$$

where μ is a measure with finite total variation. Bloomfield and Steiger (1983) and Nychka et al. (1995) considered a similar problem as that in (2.1), but used a smoothing spline penalty

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \rho_\tau(y_i - f(x_i)) + \lambda \int [f''(x)]^2 dx. \tag{2.4}$$

The minimizer of (2.4) over a second-order Sobolev space is a natural cubic spline with all design points as knots. Bosch et al. (1995) proposed an interior point algorithm which is proven to converge to solve the minimization problem.

For multi-dimensional feature space, He et al. (1998) proposed a bivariate quantile smoothing spline and He and Ng (1999) generalized the idea to multiple covariates using an ANOVA-type decomposition. Li et al. (2007) proposed a more general framework called the kernel quantile regression (KQR). By penalizing the roughness of the function estimator using its squared functional norm in a reproducing kernel Hilbert space (RKHS), the KQR solves the regularization problem

$$\min_{f\in\mathcal{H}_K} \sum_{i=1}^{n} \rho_\tau(y_i - f(\boldsymbol{x}_i)) + \frac{\lambda}{2}||f||^2_{\mathcal{H}_K}, \tag{2.5}$$

where $\mathcal{H}_K$ is a RKHS and $||\cdot||_{\mathcal{H}_K}$ is the corresponding function norm. Most recently,

Fenske et al. (2011) proposed a boosting method to select and estimate additive quantile function. Although their method was not intentionally targeting at variable selection, with moderately small number of iterations, boosting algorithm naturally achieves variable selection by using the most important predictors to update the fitted function.

Despite several existing nonparametric quantile function estimators, selecting relevant predictors in multi-dimensional data is an important yet challenging topic that has not been addressed in depth. Variable selection in quantile regression is much more difficult than that in the least square regression. The variable selection is carried at various levels of quantiles, which amounts to identifying variables that are important for the entire distribution, rather than limited to the mean function as in the least squares regression case. This has important applications to handle heteroscedastic data. Several regularization methods were proposed (Zou and Yuan, 2008a,b; Wu and Liu, 2009) for linear quantile regression. However, to our knowledge, there still lacks a method for variable selection in nonparametric quantile regression. This is the main motivation of our work.

In the presence of multiple predictors, many nonparametric estimation procedures may suffer from the curse of dimensionality. The smoothing spline analysis of variance (SS-ANOVA) model (Wahba, 1990) provides a flexible and effective estimation framework to tackle the problem. Since some of the predictors may not be useful or redundant for prediction, variable selection is important in nonparametric regression. In the context of least squares regression, the COmponent Selection and Shrinkage Operator (COSSO) (Lin and Zhang, 2006) was proposed to perform continuous function shrinkage and estimation by penalizing the sum of RKHS norms of the components. However, variable selection in nonparametric quantile regression is void in the literature. In this paper, we adopt the COSSO-type penalty to develop a new penalized framework for joint quantile estimation and variable selection. In nonparametric literature, built upon basis expansion, several methods for estimation and selection in additive models have been proposed (Meier et al., 2009; Huang et al., 2010). We prefer the COSSO penalty in a RKHS for several reasons. Despite basis expansion enjoying lower computational cost, the choice of basis functions and number of knots require further justifications. In addition, the notion of a basis expansion implicitly assumes that the true function lies in the functional space spanned by the basis functions, which is finite dimensional and should only be treated as an approximation to the true underlying functional space.

The remainder of the article is organized as follows. Section 2 reviews the SS-ANOVA models and introduces the new estimator. An iterative computation algorithm is given in Section 3, along with parameter tuning procedure. Extensive empirical studies, including both the homogeneous and heterogenous errors are given in Section 4. Three real example analysis results are presented in Section 5. We conclude our findings in Section 6.

2.2 Formulation

2.2.1 Smoothing Spline ANOVA

In the framework of smoothing spline ANOVA (SS-ANOVA), it is assumed that a function $f(\boldsymbol{x}) = f(x^{(1)}, \ldots, x^{(p)})$ has the ANOVA decomposition

$$f(\boldsymbol{x}) = b + \sum_{j=1}^{p} f_j(x^{(j)}) + \sum_{j<k} f_{j,k}(x^{(j)}, x^{(k)}) + \cdots, \tag{2.6}$$

where b is a constant, f_j's are the main effects and $f_{j,k}$'s are the two-way interactions, and so on. We estimate each of the main effects in a RKHS denoted by $\mathcal{H}_j = \{1\} \oplus \bar{\mathcal{H}}_j$ whereas the interactions are estimated in a tensor product spaces of the corresponding univariate function spaces. When $x^{(j)}$ is a continuous variable, a popular choice of $\mathcal{H}_j$ is the second-order Sobolev space $\mathcal{S}^2[0,1] = \{g : g, g' \text{ are absolutely continuous and } g'' \in \mathcal{L}^2[0,1]\}$. When endowed with the norm

$$||g||^2 = \left\{\int_0^1 g(x)dx\right\}^2 + \left\{\int_0^1 g'(x)dx\right\}^2 + \int_0^1 \{g''(x)\}^2 dx, \tag{2.7}$$

the second-order Sobolev space is a RKHS with reproducing kernel

$$R(x,y) = 1 + k_1(x)k_1(y) + k_2(x)k_2(y) - k_4(|x-y|), \tag{2.8}$$

where $k_1(x) = x - \frac{1}{2}$, $k_2(x) = \frac{1}{2}\left[k_1^4(x) - \frac{1}{12}\right]$ and $k_4(x) = \frac{1}{24}\left[k_1^4(x) - \frac{1}{2}k_1^2(x) + \frac{7}{240}\right]$. When $x^{(j)}$ is a categorical variable that takes only finite distinct values, $\{1, \ldots, L\}$, we use a different reproducing kernel

$$R(s,t) = L \cdot I(s=t) - 1, \ \ s,t \in \{1,\ldots,L\}. \tag{2.9}$$

See Wahba (1990) and Gu (2002) for more details. The entire tensor-product space for estimating $f(\boldsymbol{x})$ is given by

$$\mathcal{F} = \otimes_{j=1}^{p} \mathcal{H}_j = \{1\} \oplus \sum_{j=1}^{p} \bar{\mathcal{H}}_j \oplus \sum_{j<k} \left(\bar{\mathcal{H}}_j \otimes \bar{\mathcal{H}}_k\right) \oplus \cdots . \tag{2.10}$$

Note that $\mathcal{F} = \otimes_{j=1}^{p} \mathcal{H}_j$ is also a RKHS, and its reproducing kernel is the sum of the reproducing kernels of those component spaces.

In practice, the higher-order interactions in (2.6) will usually be truncated for convenience in interpretation and to avoid the curse of dimensionality. A general expression for a truncated space can be written as

$$\mathcal{F} = \{1\} \oplus \mathcal{F}_1 = \{1\} \oplus \left\{\oplus_{j=1}^{q} \mathcal{F}_j\right\}, \tag{2.11}$$

where $\mathcal{F}_1, \ldots, \mathcal{F}_q$ are q orthogonal subspaces of $\mathcal{F}$. A special case is the well-known additive model (Hastie and Tibshirani, 1990) with $q = p$, in which only the main effects are kept in the model, say $f(\boldsymbol{x}) = b + \sum_{j=1}^{d} f_j(x^{(j)})$. When both main effects and two-way interaction effects are retained, the truncated space has $q = p(p+1)/2$. For illustration purpose, we focus on the additive model afterward in this paper, thus all the interactions are dropped. But the idea can be naturally generalized to any function space with higher order interactions.

A typical method for estimating nonparametric quantile function is through solving the regularization problem

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - f(\boldsymbol{x}_i)) + \lambda J(f), \tag{2.12}$$

where λ is a smoothing parameter and $J(\cdot)$ is a penalty functional. A smoothing spline estimate uses the penalty $J(f) = \sum_{j=1}^{p} \theta_j^{-1} ||P^j f||^2$, where P^j is an orthogonal projection operator that projects f onto $\mathcal{F}_j$ and θ_j's are smoothing parameters. The estimation in (2.12) involves multiple tuning parameters $\theta_1, \cdots, \theta_p$, which needs to be selected properly for a good estimation results. The parameter λ is usually included and fixed at some convenient value for computational stability in practice.

2.2.2 New Methodology: COSSO-Quantile Regression

To achieve joint variable selection and function estimation in nonparametric quantile regression, we consider the following regularization problem

$$\frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i - f(\boldsymbol{x}_i)) + \lambda\sum_{j=1}^{p} w_j||P^j f||, \tag{2.13}$$

where w_j's are known weights. We will refer to (2.13) as COSSO-QR afterward.

The problem in (2.13) is a flexible modeling framework that includes several existing methods as special cases. For instance, it reduces to the L_1-norm quantile regression (Li and Zhu, 2008) in linear models. More specifically, if $f(\boldsymbol{x}) = b + \sum_{j=1}^{p}\beta_j x^{(j)}$ and we consider a linear functional space $\mathcal{F} = \{1\} \oplus \left\{\oplus_{j=1}^{p}\{x^{(j)} - 1/2\}\right\}$ with inner product $\langle f, g\rangle = \int fg$, then the RKHS norm penalty $||P^j f||$ becomes proportional to $|\beta_j|$. We allow each functional component to be penalized differently depending on its associated weight $w_j \in (0, \infty)$. In principle, smaller weights are assigned to important function components while larger weights are assigned to less important components. This is in the same spirit of the adaptive LASSO (Zou, 2006) and adaptive COSSO (Storlie et al., 2011). We propose to construct the weights w_j from the data adaptively. For each component f_j, its L_2-norm $||f_j(x)||_{L_2} = \sqrt{\int [f_j(x)]^2 dF(x)}$, $F(\cdot)$ is the distribution function of x, is a natural measure to quantify the importance of functional components. In practice, given a reasonable initial estimator $\tilde{f}$, we propose to construct the weights w's by its inverse empirical L_2-norm

$$w_j^{-1} = ||P^j\tilde{f}||_{n,L_2} = \sqrt{n^{-1}\sum_{i=1}^{n}[P^j\tilde{f}(\boldsymbol{x}_i)]^2},\ j = 1, \ldots, p. \tag{2.14}$$

A convenient choice of $\tilde{f}$ is the solution of the KQR.

Due to the fact that both the check loss and the penalty functional $J(f)$ are continuous and convex in f, the existence of the minimizer of (2.13) is guaranteed as stated in the following Theorem.

Theorem 1. *Let $\mathcal{F}$ be an RKHS of functions with the decomposition (2.11), then there exists a minimizer to (2.13) in $\mathcal{F}$.*

Directly minimizing (2.13) can be a daunting task as searching over the infinite di-

mensional space $\mathcal{F}$ for a minimizer is practically infeasible. Analogous to the smoothing spline models, the following Theorem shows that the minimizer of (2.13) lies in a finite dimensional space. This important result assures the feasibility of computation.

Theorem 2. *Representer Theorem: Let the minimizer of (2.13) be* $\hat{f} = \hat{b} + \sum_{j=1}^{p} \hat{f}_j$ *with* $\hat{f}_j \in \bar{\mathcal{H}}_j$, *then* $\hat{f}_j \in span\{R_{\mathcal{F}_j}(x_i^{(j)}, \cdot), i = 1, \ldots, n\}$ *where* $R_{\mathcal{F}_j}(\cdot, \cdot)$ *is the reproducing kernel of* $\mathcal{F}_j$.

2.3 Algorithm

To further facilitate the computation, we first present an equivalent formulation of (2.13). By introducing non-negative slack variables $\theta_j, j = 1, \ldots, p$, and using the Lemma 2 in Lin and Zhang (2006), it is easy to show that minimizing (2.13) is equivalent to solving the following optimization problem

$$\begin{aligned} \min_{f,\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - f(\boldsymbol{x}_i)) + \lambda_0 \sum_{j=1}^{p} w_j^2 \theta_j^{-1} ||P^j f||^2 \\ \text{s.t.} \quad \sum_{j=1}^{p} \theta_j \leq M, \ \theta_j \geq 0, \forall j, \end{aligned} \tag{2.15}$$

where λ_0 and M are both smoothing parameters. The roles of the slack variables θ_j's are very different from those in smoothing splines model. The slack variables θ_j's allow us to recover the sparse structure since $\theta_j = 0$ if and only if $||P^j f|| = 0$ (Lin and Zhang, 2006).

Moreover, when θ_j's are unknown, the penalty part in (2.15) reduces to that in traditional smoothing spline and thus by the Representer Theorem of Kimeldorf and Wahba (1971), the minimizer of (2.15) has the form

$$f(\boldsymbol{x}) = b + \sum_{i=1}^{n} c_i R_{\theta,w}(\boldsymbol{x}_i, \boldsymbol{x}), \tag{2.16}$$

where $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$, $b \in \mathbb{R}$, and $R_{\theta,w} = \sum_{j=1}^{p} w_j^{-2} \theta_j R_{\mathcal{F}_j}$.

Let $\boldsymbol{R}_j = \{R_{\mathcal{F}_j}(x_i^{(j)}, x_{i'}^{(j)})\}_{i,i'=1}^{n}$ be an $n \times n$ matrix and $\mathbf{1}_n$ be a column vector of n ones. When evaluated the minimizer at the design points, we write $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$ as $\boldsymbol{f} = b\mathbf{1}_n + (\sum_{j=1}^{p} w_j^{-2} \theta_j \boldsymbol{R}_j)\boldsymbol{c}$ and define $||\boldsymbol{v}||_{C_\tau} = n^{-1} \sum_{i=1}^{n} \rho_\tau(v_i)$ for a vector of length

n. The objective function in (2.15) becomes

$$\min_{b,\boldsymbol{c},\boldsymbol{\theta}} \left\| \boldsymbol{y} - b\mathbf{1}_n - \left(\sum_{j=1}^{p} \frac{\theta_j}{w_j^2} \boldsymbol{R}_j \right) \boldsymbol{c} \right\|_{C_\tau} + \lambda_0 \boldsymbol{c}^T \left(\sum_{j=1}^{p} \frac{\theta_j}{w_j^2} \boldsymbol{R}_j \right) \boldsymbol{c} \quad \text{s.t.} \quad \sum_{j=1}^{p} \theta_j \leq M,\ \theta_j \geq 0,\ \forall j. \tag{2.17}$$

For the remaining of the article, we will refer to (2.17) as the objective function of our proposed method.

2.3.1 Iterative Optimization Algorithm

It is possible to minimize the objective function in (2.17) with respect to all the parameters, $(b, \boldsymbol{c}^T, \boldsymbol{\theta}^T)^T$, simultaneously, but the programming effort can be substantial. Alternatively, we can decompose the parameters into two parts, $\boldsymbol{\theta}$ and $(b, \boldsymbol{c}^T)^T$, and then iteratively solve two sets of optimization problems in turn, with respect to $\boldsymbol{\theta}$ and $(b, \boldsymbol{c}^T)^T$. Consequently, we suggest the following iterative algorithm:

1. Fix $\boldsymbol{\theta}$, solve $(b, \boldsymbol{c}^T)^T$

$$\min_{b,\boldsymbol{c}} \left\| \boldsymbol{y} - b\mathbf{1}_n - \left(\sum_{j=1}^{p} \frac{\theta_j}{w_j^2} \boldsymbol{R}_j \right) \boldsymbol{c} \right\|_{C_\tau} + \lambda_0 \boldsymbol{c}^T \left(\sum_{j=1}^{p} \frac{\theta_j}{w_j^2} \boldsymbol{R}_j \right) \boldsymbol{c}. \tag{2.18}$$

2. Fix $(b, \boldsymbol{c}^T)^T$, solve

$$\min_{\boldsymbol{\theta}} \ \|\boldsymbol{y}^* - \boldsymbol{G}\boldsymbol{\theta}\|_{C_\tau} + \lambda_0 \boldsymbol{c}^T \boldsymbol{G} \boldsymbol{\theta},\ \text{s.t.}\ \sum_{j=1}^{p} \theta_j \leq M, \theta_j \geq 0,\ \forall j, \tag{2.19}$$

   where $\boldsymbol{y}^* = \boldsymbol{y} - b\mathbf{1}_n$ and $\boldsymbol{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_p\}$ is an $n \times p$ matrix with columns $\boldsymbol{g}_j = w_j^{-2} \boldsymbol{R}_j \boldsymbol{c}$.

The optimization problems in (2.18) and (2.19) can be cast into quadratic programming and linear programming problems, respectively. We defer all the derivations to the Appendix. So, both of them can be solved using standard optimization software, such as MATLAB and `R`.

In practice, based on our empirical experience, the algorithm converges quickly in a few steps. We have noted that the one-step solution often provides a satisfactory approximation to the solution. As a result, we advocate the use of one-step update in practice.

An important connection between our proposed method and the KQR can be unraveled by realizing that the objective function in (2.18) is exactly the same as that in the KQR. This connection suggests that when $\boldsymbol{\theta}$ is known, our proposed method shares the same spirit as the KQR. The optimization problem for estimating $\boldsymbol{\theta}$ essentially imposes the non-negative garrote (Breiman, 1995) type shrinkage on θ's, and hence achieves variable selection by shrinking some of θ_j's to zero.

2.3.2 Parameter Tuning

Like any other penalized regression problem, the performance of the new estimator critically depends on properly-tuned smoothing parameters in (2.17). Smoothing parameters play an important role in balancing the trade-off between the goodness of data fit and the model complexity. A reasonable parameter choice is usually the one that minimizes some generalized error or information criterion. In the quantile regression literature, one commonly used criterion is the Schwarz information criterion (SIC) (Schwarz, 1978; Koenker et al., 1994)

$$\log\left(\frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i-\hat{f}(\boldsymbol{x}_i))\right)+\frac{\log n}{2n}df, \tag{2.20}$$

where df is a measure of complexity of the fitted model. Various authors (Koenker, 2005; Yuan, 2006; Li et al., 2007) have argued using the number of zero residuals as an estimate of effective degrees of freedom. In our experimental study, we realized the estimated degrees of freedom fluctuates a lot among different smoothing parameters and therefore may lead to an unstable tuning result. As an alternative, we consider a bootstrap method which will be presented in the following section to estimate the degrees of freedom.

In addition to the SIC, another popular criterion to choose the smoothing parameter is k-fold cross validation, which has been widely applied to various regression and classification problems and usually gives competitive performance. In the following numerical study, we will report the result for both SIC and 5-fold cross validation.

In the following, we summarize the complete algorithm for the proposed method, including both model fitting and parameter tuning steps.

Step 1. Initialization. Set $\theta_j = 1,\ \forall j$.

Step 2. For each of the grid points of λ_0, solve (2.18) for $(b, \boldsymbol{c}^T)^T$, and record the SIC score or CV error. Choose the best λ_0 that minimizes the SIC or CV error, then fix it in later steps.

Step 3. For each of the grid points of M, solve (2.17) for $(b, \boldsymbol{c}^T, \boldsymbol{\theta}^T)^T$ using the aforementioned iterative optimization algorithm. Record the SIC score or CV error at each grid point and choose the best M that minimizes either SIC score or CV error.

Step 4. Solve (2.17) using the chosen λ_0 and M pair, on the full data. Note that this is already done if tuning was based on SIC.

Since the tuning procedure described above does not cover all the possible pairs of (λ_0, M), it would be beneficial to enhance the tuning with a refined search. In particular, we suggest to do the following. After Step 3, say, we obtain the optimal pair (λ_0^*, M^*). Then we focus on a narrowed and more focused region, the neighborhood of (λ_0^*, M^*) and apply Step2 and 3 again. The optimal parameters determined at this refined step, say, (λ_0^{**}, M^{**}) will be used as the final selection. Our simulation study also confirms that this refined tuning procedure can improve the prediction and selection performance substantially.

2.3.3 Bootstrapped Degrees of Freedom Estimate

In nonparametric quantile regression literature, using the number of zero residuals as a measure of model complexity has been widely adopted. The notion was originated from a more generic quantity, the divergence formula $\sum_{i=1}^{n} \frac{\partial \hat{f}(\boldsymbol{x}_i)}{\partial y_i}$, which first appeared in Stein's unbiased risk estimation (SURE) (Stein, 1981) and later been extensively used to evaluate the effective degrees of freedom for various modeling procedures, see Ye (1998), Efron (2004), Koenker (2005) and reference therein.

In our pilot study, we realized that the number of zero residuals fluctuate a lot across a wide range of smoothing parameters in our COSSO-QR method. Hence we do not think it is a reliable estimate. Without an informative measure of the model complexity, SIC can not be an effective tuning procedure. To alleviate the unstable estimate, we consider directly estimating the derivative using bootstrap. The procedure can be summarized into following steps.

1. For a particular pair of smoothing parameters (λ_0, M), fit the COSSO-QR model in equation (2.17) and store the fitted values $\hat{f}_i$ and residuals $r_i = y_i - \hat{f}_i$.

2. Repeat this step B times.

   2.1. Generate a bootstrapped response

   $$y_i^{boot} = \hat{f}_i + r_i^{boot},\ r_i^{boot} \overset{iid}{\sim} \hat{F}_n(r),\ i = 1, \ldots, n, \tag{2.21}$$

   where $\hat{F}_n$ is the empirical distribution function of the residuals.

   2.2. Fit the COSSO-QR model but replace the original response by the bootstrapped response and record the fitted values $\hat{f}_i^{boot}$.

3. For each $i = 1, \ldots, n$, fit a simple linear regression model by regressing $\hat{f}_i^{boot}$ on y_i^{boot} and use the estimated slope as an estimate of the derivative. Thus the bootstrapped degrees of freedom estimate is given by summing up the n estimated slopes.

Estimating the derivative by regression slope was pioneered in Ye (1998). To better suit our quantile regression model, we generate perturbed response by bootstrapping rather than adding artificial noise to the observed response as used in Ye's original proposal. As remarked by Ye (1998), different perturbation methods have minor influence on estimating the slope and will not lead to considerable bias.

2.4 Numerical Results

In this section we present the empirical performance of the COSSO-QR procedure using simulated data. For the experiment design, we use the following functions as building blocks: $g_1(t) = t$; $g_2(t) = (2t-1)^2$; $g_3(t) = \frac{\sin(2\pi t)}{2-\sin(2\pi t)}$ and $g_4(t) = 0.1\sin(2\pi t) + 0.2\cos(2\pi t) + 0.3\sin^2(2\pi t) + 0.4\cos^3(2\pi t) + 0.5\sin^3(2\pi t)$. Similar settings were also considered in Lin and Zhang (2006).

We evaluate the performance from two aspects, prediction accuracy and model selection. The integrated absolute error (IAE), defined as $\text{IAE} = E|\hat{f}(\mathbf{X}) - f(\mathbf{X})|$, is used to assess prediction accuracy, where the expectation is evaluated by a monte carlo integration with 10,000 test points generated from the same distribution as the training data. In terms of model selection, we first denote $\hat{\mathcal{M}} = \{j : \hat{\theta}_j \neq 0\}$ and $\mathcal{M}_0 = \{j : ||P^j f|| > 0\}$

as the selected model and true model, respectively, and $|\mathcal{M}|$ as the cardinality of the set $\mathcal{M}$. Then we compute four statistics for assessing selection accuracy: correct selection, $I(\hat{\mathcal{M}} = \mathcal{M}_0$, type I error rate, $\frac{|\hat{\mathcal{M}} \cap \mathcal{M}_0^c|}{p-|\mathcal{M}_0|}$, power, $\frac{|\hat{\mathcal{M}} \cap \mathcal{M}_0|}{|\mathcal{M}_0|}$, and model size, $|\hat{\mathcal{M}}|$. For the purpose of comparison, we also include the solution of the KQR fitted with only relevant predictors based on 5-fold cross validation tuning. This method will later be referred to as the Oracle estimator. The Oracle estimator provides a benchmark for the best possible estimation risk if the important variables were known. We also include two existing methods, the KQR and boosting QR (Fenske et al., 2011), for comparisons.

Another property that we would like to study is the role of the adaptive weights in the performance of the COSSO-QR procedure. Without any a priori knowledge on the importance of each predictor, we can set all $w_j = 1$ in (2.13) and proceed to solve the objective function. For the adaptive proposal, we use the KQR as an initial estimate, $\tilde{f}$, to produce an adaptive weight.

Three different quantile values $\tau = \{0.2, 0.5, 0.8\}$, are used throughout the simulation. For each of the following examples, we repeat 100 times and report the average summary statistics and their associated standard errors.

We consider two simulation models with detailed result given in Tables 2.2 to 2.5.

2.4.1 Computational Cost

Before introducing simulation models, we first study how computationally intense our method is. To assess the computational cost, we present the average elapsed CPU times for solving equation (2.15) for a fixed pair of (λ_0, M) over 200 replicates. The predictors $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(p)})$ are independently generated from $U(0, 1)$ and then take n observations from the model

$$y_i = 5g_1(x_i^{(1)}) + 3g_2(x_i^{(2)}) + 4g_3(x_i^{(7)}) + 6g_4(x_i^{(10)}) + \varepsilon, \ i = 1, \ldots, n, \tag{2.22}$$

where ε_i are independently drawn from $t(3)$. We consider multiple sample size n and dimension p combinations. All computations are done on a desktop PC with an Intel Core i7-2600K CPU and 12GB of memory. The average CPU times are summarized in Table 2.1.

According to the algorithm we provided in 2.3.2, it is understandable that most of the computational cost comes from solving the quadratic programming in (2.18). Hence,

the computing time substantially increases from $n = 100$ to $n = 300$, but varies little between different number of inputs and quantiles.

2.4.2 Homoskedastic Error Model

We first consider generating response from a location family given in (2.22) and keep the error distribution unchanged. It follows that the $100\tau\%$ quantile function in the homoskedastic model is given by

$$Q_\tau(y|\boldsymbol{x}) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(7)}) + 6g_4(x^{(10)}) + F_\varepsilon^{-1}(\tau), \tag{2.23}$$

where $F_\varepsilon(\cdot)$ is the distribution function of ε. To examine the model selection of COSSO-QR, we generate predictors $x^{(j)},\ j = 1, \ldots, 40$, marginally from $U(0, 1)$ and consider an autoregressive type of dependency by letting pairwise correlation $\text{cor}(x^{(j)}, x^{(k)}) = \rho^{|j-k|},\ \forall j \neq k$. Two levels of dependency will be used $\rho = \{0, 0.7\}$. We use sample size $n = 200$ in this case and present the performance of five competing procedures: KQR, boosting QR, COSSO-QR, adaptive COSSO-QR, and the Oracle estimator, in Table 2.2 and 2.3.

Another interesting observation can be made by examining the robustness property of the COSSO-QR in estimating the conditional median function. Although least squares regression and quantile regression are not generally comparable, the conditional median and mean functions coincide in this example, making the comparison between them justifiable. Thus, we incorporate two sparse least squares nonparametric regression models, COSSO (Lin and Zhang, 2006) and adaptive COSSO (Storlie et al., 2011), to estimate the conditional mean function and see how our method compare with them.

From Table 2.2 and 2.3, in terms of prediction error, the adaptive COSSO-QR has the smallest IAE, which is quite close to that of the Oracle, and is hence the best, followed by COSSO-QR, boosting QR, and the KQR is the worst. Besides, using 5-fold CV produces better estimation result than SIC most of the times. It is clear that the KQR suffers considerably from those noisy variables. Although boosting QR has size greater than 28, the four important predictors are selected most of the time during the boosting update, whereas the other noise variables are chosen much less frequently and hence the performance is not severely affected by them. Fenske et al. (2011) also found similar result and recommended using the frequency a predictor is chosen as a guidance

for variable selection.

With regard to variable selection, the proposed COSSO-QR is effective in identifying important variables and removing noise variables, particularly when using SIC as a tuning procedure, which is shown by its small Type I error and large power. The adaptive COSSO-RQ tends to select a slightly larger model size, thus increases both Type I error and power. Overall speaking, the COSSO-QR procedure shows promising performance in terms of both variable selection and quantile estimation in this example.

The conditional mean function estimators, COSSO and adaptive COSSO, give comparable model selection. However, as can be seen from IAE, their estimations are seriously affected by the heavy tail of the error distribution. Benefited from their sparse property, they still perform better than KQR but are less competitive to the other procedures, even with adaptive weight. In addition, the standard errors are almost 10 tens larger than the other median estimators, implying our COSSO-QR method enjoys the robust property when median is of interest.

Figure 1 gives a graphical illustration for the fitted curve and pointwise confidence band given by the adaptive COSSO-QR for $\tau = 0.2$. For comparison, the estimated functions by the Oracle are also depicted. We apply each procedure to 100 simulated datasets and a pointwise confidence band is given by the 5% and 95% percentiles. Figure 1 suggests that the COSSO-QR produces a very good estimation for the true functions, and the fits are comparable to those given by the Oracle estimator. The fourth function component is more difficult to estimate due to its subtle features in extreme values and inflexion points.

When predictors are correlated, Table 2.3 shows our procedure is slightly affected. Overall, the type I error is well-controlled within 5% and the power is close to 90% most of the time. Moreover, the IAE suggests we do not lose too much efficiency relative to the Oracle estimator.

2.4.3 Heteroskedastic Error Model

To further examine the finite sample performance of the new methods, we consider generating response from a location-scale family

$$y_i = 5g_1(x_i^{(1)}) + 3g_2(x_i^{(2)}) + 4g_3(x_i^{(7)}) + 6g_4(x_i^{(10)}) + \exp\left[2g_3(x_i^{(12)})\right]\varepsilon,\ i = 1, \ldots, n, \quad (2.24)$$

where $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$. In the heteroskedastic model, the $100\tau\%$ quantile function is given by

$$Q_\tau(y|\boldsymbol{x}) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(7)}) + 6g_4(x^{(10)}) + \exp\left[2g_3(x^{(12)})\right]\Phi^{-1}(\tau), \quad (2.25)$$

where $\Phi(\cdot)$ is the distribution function of standard normal. The predictors are generated in the same fashion as that in the previous example and we use sample size $n = 300$ in this case.

From this example, we aim to evaluate the performance of the COSSO-QR under the scenario where some variables can only be influential on a certain range of quantiles. More specifically, like the homoskedastic example, the median function depends on the 1, 2, 7 and 10th predictors. However, other than the median function, $x^{(12)}$ will not only be influential but its effect becomes larger and larger toward the tails.

Table 2.4 and 2.5 summarize the performance of all competing methods. Again, we observe that the adaptive COSSO-QR performs the best and the KQR is the worst in terms of prediction error. However, the boosting QR provides as efficient or sometimes even more efficient estimation than the COSSO-QR. Similarly, for variable selection, by penalizing a more complicated model, SIC tuning procedure usually selects a smaller model size and identifies the correct model more frequently.

Another point we would like to emphasize is that when $\tau = 0.5$, the estimated model size is close to 4 as expected, since the median only depends on the 1,4,7 and 10th predictors. When τ is away from 0.5, our COSSO-QR procedures can successfully identify the additional informative predictor in the error variance, suggesting that the new method's capability to identify all the relevant predictors that influence the distribution of the response.

2.5 Real Data Analysis

We apply the COSSO-QR method to two real datasets: prostate cancer data and ozone data. The prostate data is from Stamey et al. (1989), consisting of 97 patients who were about to receive a radical prostatectomy. This data was used by Tibshirani (1996) to model the mean function of the level of prostate-specific antigen on 8 clinical outcomes and select relevant variables. The ozone data contains 330 observations collected in Los Angeles in 1976, and the purpose of the study is to model the relationship between the

daily ozone concentration and 8 meteorological covariates. The data has been used in various studies (Buja et al., 1989; Breiman, 1995; Lin and Zhang, 2006). These two data are publicly available in the `R` packages `ElemStatLearn` and `cosso`, respectively.

We apply our methods on these datasets and estimate the prediction risk, $E\rho_\tau(Y - f(\boldsymbol{X}))$, by randomly reserving 10% of the data as testing set. The smoothing parameters, tuned by 5-fold cross validation, and model parameters are estimated using only the training set. The estimated parameters will then be applied on the testing set and the prediction risk is used as a comparison between various methods. The entire procedure is repeated 100 times and averaged.

Table 2.6 summarizes the prediction risk along with its associated standard error. Based on the result, the adaptive weights is not always helpful in real application. The advantage of adaptive weight is more perceivable in the prostate data. But, with or without adaptive weight, the differences between them are usually within reasonable error margin. Overall, the key observation is that our proposed method provides competitive and usually superior prediction than the existing methods.

Apart from comparing prediction error, we also apply our methods to the complete prostate data and summarize variable selection. An interesting comparison is that in the study of mean function, Tibshirani (1996) selected three prominent predictors, log-cancer volume, log-weight and seminal vesicle invasion. These three predictors are also selected by our approach when we consider the median. However, in the 20% quantile, gleason score shows up as an additional predictor. Meanwhile, in the 80% quantile, only two predictors are chosen, log-cancer volume and seminal vesicle invasion, but not log-weight.

2.6 Conclusions

We propose a new regularization method that simultaneously selects important predictors and estimate the conditional quantile function. Our method is available in the `R` package `cosso` version 2.0-2. The COSSO-QR method conquers the limitation of selecting only predictors that influence the conditional mean in least square regression, facilitating the analysis of the full conditional distribution. The proposed method also includes the L_1-norm quantile regression and the KQR as special cases. In a simulation study and real data analysis, our method provides satisfactory model fitting and great potential for

selecting important predictors.

The number of predictors we consider in both simulation and real data is moderate. With advancement of modern technology, high-throughput data becomes more frequent nowadays. In ultra-high dimensional feature space, Fan et al. (2011) recently proposed a screening procedure for nonparametric regression model. Further study can work toward incorporating a suitable screening procedure as a first step and then apply our proposed method at the second in a ultra-high dimensional feature space.

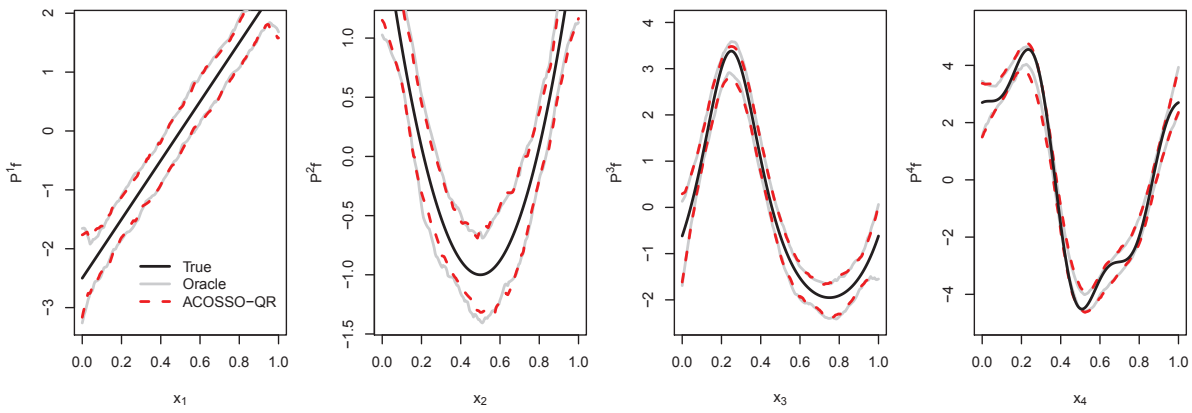

Figure 2.1: The fitted function components and their associated pointwise confidence band for homoskedastic example with independent features. The dark solid line is for the true function component, the light solid line is for the Oracle estimator and the broken line is for the adaptive COSSO-QR estimator.

Table 2.1: Elapsed CPU time (in seconds) for solving COSSO-QR model.

	(n, p)					
τ	(100,10)	(100,40)	(200,10)	(200,40)	(300,10)	(300,40)
0.2	0.041	0.045	0.300	0.329	0.998	1.154
0.5	0.038	0.044	0.278	0.299	0.914	1.048

Table 2.2: Simulation results for the homoskedasitc example with independent features. The standard errors are given in the parentheses.

τ	Method	Correct	Type I Error	Power	Model Size	IAE
	KQR	-	-	-	-	2.223 (0.019)
	Boosting QR	0.00 (0.00)	0.67 (0.02)	1.00 (0.00)	28.13 (0.60)	1.098 (0.016)
	COSSO-QR-5CV	0.70 (0.05)	0.01 (0.00)	0.98 (0.01)	4.23 (0.09)	0.949 (0.021)
0.2	COSSO-QR-SIC	0.81 (0.04)	0.02 (0.01)	0.99 (0.01)	4.58 (0.18)	0.983 (0.021)
	ACOSSO-QR-5CV	0.73 (0.05)	0.02 (0.01)	0.99 (0.01)	4.69 (0.20)	0.645 (0.016)
	ACOSSO-QR-SIC	0.82 (0.04)	0.02 (0.00)	1.00 (0.00)	4.53 (0.15)	0.667 (0.016)
	Oracle	-	-	-	-	0.634 (0.011)
	KQR	-	-	-	-	1.921 (0.017)
	Boosting QR	0.00 (0.00)	0.76 (0.02)	1.00 (0.00)	31.18 (0.60)	0.781 (0.009)
	COSSO-QR-5CV	0.84 (0.04)	0.01 (0.00)	1.00 (0.00)	4.36 (0.15)	0.612 (0.015)
	COSSO-QR-SIC	0.92 (0.03)	0.00 (0.00)	0.99 (0.01)	4.08 (0.13)	0.638 (0.017)
0.5	ACOSSO-QR-5CV	0.82 (0.04)	0.01 (0.00)	1.00 (0.00)	4.39 (0.11)	0.461 (0.008)
	ACOSSO-QR-SIC	0.93 (0.03)	0.00 (0.00)	0.99 (0.00)	4.07 (0.06)	0.505 (0.012)
	COSSO	0.83 (0.04)	0.01 (0.00)	0.99 (0.01)	4.15 (0.05)	0.824 (0.025)
	ACOSSO	0.76 (0.04)	0.01 (0.00)	0.99 (0.01)	4.30 (0.08)	0.616 (0.017)
	Oracle	-	-	-	-	0.489 (0.007)
	KQR	-	-	-	-	2.269 (0.021)
	Boosting QR	0.00 (0.00)	0.68 (0.02)	1.00 (0.00)	28.43 (0.55)	0.978 (0.014)
	COSSO-QR-5CV	0.69 (0.05)	0.01 (0.00)	0.97 (0.01)	4.31 (0.11)	0.904 (0.022)
0.8	COSSO-QR-SIC	0.78 (0.04)	0.02 (0.01)	0.98 (0.01)	4.68 (0.23)	0.944 (0.024)
	ACOSSO-QR-5CV	0.74 (0.04)	0.02 (0.01)	0.99 (0.00)	4.60 (0.17)	0.661 (0.015)
	ACOSSO-QR-SIC	0.88 (0.03)	0.01 (0.00)	0.99 (0.00)	4.30 (0.15)	0.726 (0.017)
	Oracle	-	-	-	-	0.644 (0.011)

Table 2.3: Simulation results for the homoskedasitc example with dependent features. The standard errors are given in the parentheses.

τ	Method	Correct	Type I Error	Power	Model Size	IAE
	KQR	-	-	-	-	1.743 (0.015)
	Boosting QR	0.00 (0.00)	0.54 (0.02)	1.00 (0.00)	23.50 (0.73)	0.992 (0.020)
	COSSO-QR-5CV	0.22 (0.04)	0.03 (0.00)	0.87 (0.01)	4.39 (0.15)	0.935 (0.017)
0.2	COSSO-QR-SIC	0.18 (0.04)	0.02 (0.01)	0.84 (0.01)	4.22 (0.21)	0.978 (0.018)
	ACOSSO-QR-5CV	0.23 (0.04)	0.03 (0.01)	0.88 (0.01)	4.52 (0.23)	0.690 (0.014)
	ACOSSO-QR-SIC	0.23 (0.04)	0.03 (0.01)	0.87 (0.01)	4.67 (0.28)	0.710 (0.014)
	Oracle	-	-	-	-	0.609 (0.011)
	KQR	-	-	-	-	1.512 (0.012)
	Boosting QR	0.00 (0.00)	0.50 (0.02)	1.00 (0.00)	21.97 (0.76)	0.700 (0.010)
	COSSO-QR-5CV	0.18 (0.04)	0.04 (0.01)	0.89 (0.01)	4.93 (0.23)	0.718 (0.014)
	COSSO-QR-SIC	0.27 (0.05)	0.03 (0.01)	0.90 (0.01)	4.83 (0.22)	0.711 (0.015)
0.5	ACOSSO-QR-5CV	0.38 (0.05)	0.04 (0.01)	0.96 (0.01)	5.23 (0.23)	0.488 (0.011)
	ACOSSO-QR-SIC	0.60 (0.05)	0.02 (0.00)	0.95 (0.01)	4.51 (0.16)	0.481 (0.012)
	COSSO	0.35 (0.05)	0.01 (0.00)	0.88 (0.01)	3.90 (0.09)	1.436 (0.103)
	ACOSSO	0.39 (0.05)	0.01 (0.00)	0.91 (0.01)	4.13 (0.10)	0.780 (0.088)
	Oracle	-	-	-	-	0.459 (0.007)
	KQR	-	-	-	-	1.700 (0.014)
	Boosting QR	0.00 (0.00)	0.45 (0.02)	1.00 (0.00)	20.36 (0.75)	0.954 (0.016)
	COSSO-QR-5CV	0.09 (0.03)	0.04 (0.01)	0.84 (0.01)	4.83 (0.20)	0.979 (0.016)
0.8	COSSO-QR-SIC	0.16 (0.04)	0.06 (0.01)	0.90 (0.01)	5.84 (0.25)	0.971 (0.016)
	ACOSSO-QR-5CV	0.12 (0.03)	0.06 (0.01)	0.89 (0.01)	5.54 (0.29)	0.731 (0.017)
	ACOSSO-QR-SIC	0.27 (0.04)	0.05 (0.01)	0.92 (0.01)	5.52 (0.22)	0.723 (0.017)
	Oracle	-	-	-	-	0.592 (0.011)

Table 2.4: Simulation results for the heteroskedasitc example with independent features. The standard errors are given in the parentheses.

τ	Method	Correct	Type I Error	Power	Model Size	IAE
	KQR	-	-	-	-	2.422 (0.019)
	Boosting QR	0.00 (0.00)	0.75 (0.01)	1.00 (0.00)	30.72 (0.56)	1.289 (0.017)
	COSSO-QR-5CV	0.34 (0.05)	0.01 (0.00)	0.91 (0.01)	5.04 (0.11)	1.419 (0.028)
0.2	COSSO-QR-SIC	0.43 (0.05)	0.01 (0.00)	0.92 (0.01)	4.92 (0.11)	1.474 (0.026)
	ACOSSO-QR-5CV	0.57 (0.05)	0.02 (0.00)	0.98 (0.01)	5.65 (0.13)	0.976 (0.026)
	ACOSSO-QR-SIC	0.65 (0.05)	0.00 (0.00)	0.94 (0.01)	4.84 (0.10)	1.154 (0.027)
	Oracle	-	-	-	-	0.663 (0.013)
	KQR	-	-	-	-	1.718 (0.015)
	Boosting QR	0.00 (0.00)	0.74 (0.01)	1.00 (0.00)	30.63 (0.41)	0.681 (0.009)
	COSSO-QR-5CV	0.93 (0.03)	0.00 (0.00)	1.00 (0.00)	4.09 (0.05)	0.538 (0.013)
0.5	COSSO-QR-SIC	0.97 (0.02)	0.00 (0.00)	1.00 (0.00)	3.97 (0.02)	0.569 (0.016)
	ACOSSO-QR-5CV	0.84 (0.04)	0.01 (0.00)	1.00 (0.00)	4.26 (0.07)	0.371 (0.007)
	ACOSSO-QR-SIC	0.95 (0.02)	0.00 (0.00)	1.00 (0.00)	4.11 (0.07)	0.390 (0.010)
	Oracle	-	-	-	-	0.391 (0.005)
	KQR	-	-	-	-	2.446 (0.024)
	Boosting QR	0.00 (0.00)	0.74 (0.02)	1.00 (0.00)	30.72 (0.56)	1.223 (0.017)
	COSSO-QR-5CV	0.53 (0.05)	0.02 (0.00)	0.96 (0.01)	5.51 (0.16)	1.194 (0.031)
0.8	COSSO-QR-SIC	0.65 (0.05)	0.01 (0.00)	0.94 (0.01)	4.98 (0.12)	1.336 (0.029)
	ACOSSO-QR-5CV	0.60 (0.05)	0.03 (0.01)	0.99 (0.00)	6.04 (0.20)	0.923 (0.024)
	ACOSSO-QR-SIC	0.76 (0.04)	0.01 (0.00)	0.97 (0.01)	5.18 (0.13)	1.090 (0.022)
	Oracle	-	-	-	-	0.651 (0.013)

Table 2.5: Simulation results for the heteroskedasitc example with dependent features. The standard errors are given in the parentheses.

τ	Method	Correct	Type I Error	Power	Model Size	IAE
0.2	KQR	-	-	-	-	1.735 (0.018)
	Boosting QR	0.00 (0.00)	0.78 (0.02)	1.00 (0.00)	32.06 (0.55)	1.267 (0.012)
	COSSO-QR-5CV	0.11 (0.03)	0.07 (0.01)	0.91 (0.01)	7.02 (0.28)	1.155 (0.023)
	COSSO-QR-SIC	0.19 (0.04)	0.05 (0.01)	0.90 (0.01)	6.35 (0.27)	1.188 (0.022)
	ACOSSO-QR-5CV	0.21 (0.04)	0.08 (0.01)	0.97 (0.01)	7.46 (0.31)	0.822 (0.020)
	ACOSSO-QR-SIC	0.38 (0.05)	0.03 (0.01)	0.94 (0.01)	5.84 (0.23)	0.891 (0.022)
	Oracle	-	-	-	-	0.613 (0.013)
0.5	KQR	-	-	-	-	1.343 (0.012)
	Boosting QR	0.00 (0.00)	0.73 (0.00)	1.00 (0.00)	30.38 (0.15)	0.562 (0.004)
	COSSO-QR-5CV	0.41 (0.05)	0.04 (0.01)	0.95 (0.00)	5.11 (0.18)	0.653 (0.016)
	COSSO-QR-SIC	0.51 (0.05)	0.03 (0.01)	0.95 (0.00)	4.87 (0.24)	0.656 (0.014)
	ACOSSO-QR-5CV	0.45 (0.05)	0.05 (0.01)	0.99 (0.00)	5.68 (0.27)	0.413 (0.012)
	ACOSSO-QR-SIC	0.70 (0.05)	0.03 (0.01)	0.99 (0.00)	4.86 (0.26)	0.405 (0.011)
	Oracle	-	-	-	-	0.383 (0.006)
0.8	KQR	-	-	-	-	1.731 (0.018)
	Boosting QR	0.00 (0.00)	0.65 (0.01)	1.00 (0.00)	27.60 (0.50)	1.047 (0.003)
	COSSO-QR-5CV	0.19 (0.04)	0.06 (0.01)	0.90 (0.01)	6.64 (0.24)	1.111 (0.026)
	COSSO-QR-SIC	0.25 (0.04)	0.04 (0.01)	0.88 (0.01)	5.64 (0.21)	1.139 (0.023)
	ACOSSO-QR-5CV	0.41 (0.05)	0.06 (0.01)	0.98 (0.01)	6.85 (0.28)	0.796 (0.021)
	ACOSSO-QR-SIC	0.61 (0.05)	0.02 (0.00)	0.96 (0.01)	5.31 (0.13)	0.839 (0.023)
	Oracle	-	-	-	-	0.629 (0.015)

Table 2.6: Estimated prediction risk for real data. The standard errors are given in the parentheses.

		Methods			
τ	Data	KQR	Boosting QR	COSSO-QR	ACOSSO-QR
0.2	Prostate	0.261 (0.009)	0.316 (0.022)	0.232 (0.007)	0.228 (0.006)
	Ozone	1.115 (0.007)	1.136 (0.018)	1.093 (0.016)	1.096 (0.016)
0.5	Prostate	0.333 (0.011)	0.350 (0.016)	0.293 (0.008)	0.294 (0.007)
	Ozone	1.629 (0.022)	1.662 (0.018)	1.632 (0.023)	1.656 (0.024)
0.8	Prostate	0.355 (0.013)	0.280 (0.011)	0.213 (0.005)	0.205 (0.005)
	Ozone	1.160 (0.017)	1.161 (0.019)	1.156 (0.017)	1.179 (0.017)

Chapter 3

A unified variable selection and function estimation procedure for nonparametric regression using least squares approximation

3.1 Introduction

Traditional modeling procedures usually involve two steps. In the initial model building step, a large number of predictors are kept to avoid possible modeling bias, then a variable selection procedure is carried out in the second step for better interpretation and prediction (Fan and Li, 2001). More recently, the boundary between these two steps has gradual vanished. Efficient modeling procedures that perform joint estimation and selection are now mainstream in modern statistics. In addition, in the age of data deluge, more and more variables can be collected at the same time thanks to the advancement in data collection. To find the "needles" in the "haystack", the research trajectory of modern statistical modeling has headed toward variable selection.

In the nonparametric context, variable selection is a challenging task for several reasons, partly because of the difficulty of infinite dimensionality of the functional space. Traditionally, variable selection in nonparametric regression is done by borrowing some concepts developed in parametric models. For instance, an analogous stepwise or greedy search algorithm is used in CART (Breiman et al., 1984), TURBO (Friedman and Sil-

verman, 1989) and MARS (Friedman, 1991). However, based on what we learned from parametric models, discrete selection procedures, such as stepwise and greedy search, are known to suffer from several limitations (Breiman, 1995; Efron et al., 2004). From a nonparametric perspective, Yau et al. (2003) developed a Bayesian method for variable selection and Zhang et al. (2004) proposed a likelihood basis pursuit method for variable selection and estimation for exponential family.

Since the ground-breaking work of LASSO (Tibshirani, 1996, 2011), a considerable amount of work on statistical methods are centered around penalized regression. In parametric models, penalized methods, such as LASSO, are appealing for their shrinkage property by setting small coefficients to exact zeros, so that selection and estimation can be done simultaneously. In nonparametric modeling, Lin and Zhang (2006) proposed the COmponent Selection and Smoothing Operator (COSSO) penalty in the smoothing spline analysis of variance (SS-ANOVA) framework. By showing that COSSO can be viewed as an extension of the LASSO penalty to nonparametric models, the COSSO penalty naturally inherits some desirable properties. For instance, it successfully addresses the stability issue of the aforementioned discrete types of selection methods by imposing a soft-thresholding operator to achieve continuous selection. Moreover, Storlie et al. (2011) later introduced an adaptive weight that allows a different amount of shrinkage for each functional component and proved the adaptive COSSO penalty enjoys a nonparametric Oracle property under certain assumptions.

The theoretical properties of (A)COSSO have been studied exclusively in the least squares context, but rather less is known about them for other regression models, such as generalized linear models (GLM), quantile regression models and Cox proportional hazard models. Numerically, (A)COSSO exhibits promising performance in terms of selection and estimation in various regression models (Zhang and Lin, 2006; Leng and Zhang, 2007), suggesting similar theoretical results should sustain beyond the least squares context. However, such tasks can be daunting and need to be done in a case-by-case basis. In this work, we aim to explore the possibility of providing a unified framework to study the COSSO-type methods.

We motivate this work from the successful development of least squares approximation (LSA) in the parametric model (Wang and Leng, 2007). Theoretically, the LSA provides an asymptotic equivalent framework for various LASSO-types of problems and hence the asymptotic results can be established in a unified fashion. Computationally,

the LSA requires an initial parameter estimate and its associated covariance matrix as inputs and then solve an adaptive LASSO problem. Both initial estimator and adaptive LASSO problems can be solved very efficiently. The success of LSA stimulates us to explore its applicability in nonparametric model in terms of both numerical and theoretical perspectives.

In this study, we propose a nonparametric least squares approximation (NPLSA) method for a unified COSSO estimation. Parallel to the parametric LSA, the proposed method is also a two-stage method that consists of an initial estimation step and a final selection and estimation step. We introduce the NPLSA and the two-stage method in Section 2. Section 3 presents the effectiveness of NPLSA using both simulated and real examples and we conclude the article with some discussion in Section 4.

3.2 Nonparametric Least Squares Approximation

3.2.1 Model and Notations

Let $\{(y_i, \boldsymbol{x}_i) : i = 1, \ldots, n\}$ be a random sample, where y_i is the response and $\boldsymbol{x}_i = (x_i^{(1)}, \ldots, x_i^{(p)})$ is the p-dimensional predictors. Denote $f_0(\boldsymbol{x})$ as an underlying function that relates some aspect of the predictors to the response. For example, $f_0(\boldsymbol{x})$ can be the conditional mean, conditional quantile, etc. A typical nonparametric estimator of $f_0(\boldsymbol{x})$ is the minimizer of a loss function, which may be the negative of a likelihood function, plus a penalty term. We formulate the problem in a generic form

$$\min_{f \in \mathcal{F}} \mathcal{L}_n(f) + \frac{\lambda}{2} J(f), \tag{3.1}$$

where $\mathcal{F}$ is a structured functional space, $\mathcal{L}_n$ is a reasonable loss or negative likelihood function, $J(f)$ is the roughness penalty of the function f and λ is a smoothing parameter that governs the goodness-of-fit and function complexity.

The penalty functional $J(\cdot)$ is required to avoid interpolation in (3.1) for nonparametric methods to control the smoothness of the estimate function. A common choice of penalty functional is the sum of individual squared functional norm in a reproducing kernel Hilbert space (RKHS) which we will address later. For simultaneous estimation and selection, a direct sparse function estimator can be obtained by using the COSSO penalty for $J(\cdot)$, which penalizes the sum of the RKHS norms. Although a direct sparse

estimator is conceptually available for various regression models, a tailored computational algorithm is needed. One key motivation of this study is to provide a unified selection and estimation framework.

3.2.2 Least Squares Approximation

The LSA in parametric model was motivated from a simple Taylor expansion. In parametric model, the loss function is a function of unknown parameters $\boldsymbol{\beta}$. Suppose the loss function has continuous second-order derivative, then the Taylor expansion of the loss function around an initial estimate $\tilde{\boldsymbol{\beta}}$ is given by

$$\mathcal{L}_n(\boldsymbol{\beta}) \approx \mathcal{L}_n(\tilde{\boldsymbol{\beta}}) + \dot{\mathcal{L}}_n(\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \frac{1}{2}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T \ddot{\mathcal{L}}_n(\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}), \tag{3.2}$$

where $\dot{\mathcal{L}}_n$ is a gradient vector of $\mathcal{L}_n$ with respect to $\boldsymbol{\beta}$ and $\ddot{\mathcal{L}}_n$ is a Hessian matrix. When evaluated at the initial estimate $\tilde{\boldsymbol{\beta}}$, the gradient $\dot{\mathcal{L}}_n$ is a zero vector and $\mathcal{L}_n$ is a constant independent of $\boldsymbol{\beta}$. So, the only part that involves unknown parameter $\boldsymbol{\beta}$ is the quadratic term, which gives rise to the name least squares approximation.

Let $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$ be a vector of length n containing the values of an arbitrary function f evaluated at the design points. Analogous to the quadratic term in parametric LSA, we consider a squared approximation to the loss function $\mathcal{L}_n$ in (3.1) by

$$\mathcal{L}_n(\boldsymbol{f}) \approx \frac{1}{2}(\tilde{\boldsymbol{f}} - \boldsymbol{f})^T \mathrm{cov}^{-1}(\tilde{\boldsymbol{f}})(\tilde{\boldsymbol{f}} - \boldsymbol{f}). \tag{3.3}$$

In practice, we can derive the diagonal elements of the covariance matrix of $\tilde{\boldsymbol{f}}$, but the complete covariance matrix is not always available. In addition, even if the covariance matrix is available, it will not be invertible in general. To give more insight into the structure of the covariance matrix, we use a smoothing spline model as an example. Since the smoothing spline model is a linear smoother, we can write $\tilde{\boldsymbol{f}} = \boldsymbol{S}\boldsymbol{y}$, where $\boldsymbol{S}$ is the smoother matrix. Moreover, the covariance matrix of $\tilde{\boldsymbol{f}}$ is given by $\mathrm{cov}(\tilde{\boldsymbol{f}}) = \sigma^2 \boldsymbol{S}\boldsymbol{S}^T$, assuming y_i's are *i.i.d.* random sample with constant variance σ^2. The difficulty is, however, $\boldsymbol{S}$ is not full rank (Gu, 2002), and thus neither $\boldsymbol{S}$ nor $\boldsymbol{S}\boldsymbol{S}^T$ is invertible. Due to the rank deficiency, we replace $\mathrm{cov}(\tilde{\boldsymbol{f}})$ by its diagonal elements and denote it as $\boldsymbol{V} = \mathrm{diag}(v_1, \ldots, v_n)$ afterward.

3.2.3 Initial Estimation

The main difficulty of estimating a multivariate function through solving equation (3.1) is the curse of dimensionality. A popular strategy for effective and flexible multivariate function estimation is the SS-ANOVA models (Wahba, 1990). In the SS-ANOVA framework, a multivariate function is decomposed into a constant term, main effects and interactions, see Wahba (1990), Gu (2002) and Wang (2011) for a detailed discussion. To control the model complexity and avoid the curse of dimensionality, higher-order interactions are usually truncated. In this article, for convenience, we will discuss the case when main effects are kept while all the interactions are dropped, which is commonly referred to as an additive model (Hastie and Tibshirani, 1990). The general case follows with just an increase in notational complexity. In an additive model, the decomposition of a multivariate function f is given by

$$f(\boldsymbol{x}) = b + \sum_{j=1}^{p} f_j(x^{(j)}), \tag{3.4}$$

and each function f_j is estimated within a functional space $\mathcal{F}_j$. Thus, the functional space we construct to estimate f is given by

$$\mathcal{F} = \{1\} \oplus \left\{ \oplus_{j=1}^{p} \mathcal{F}_j \right\}, \tag{3.5}$$

where each functional space $\mathcal{F}_j$ is orthogonal to the constant functional space $\{1\}$ for identifiability.

When $x^{(j)}$ is a continuous variable, a popular choice of $\mathcal{F}_j$ is the second-order Sobolev space $\mathcal{S}^2[0,1] = \{g : g, g' \text{ are absolutely continuous and } g'' \in \mathcal{L}^2[0,1]\}$. When endowed with the norm

$$||g||^2 = \left\{ \int_0^1 g(x)dx \right\}^2 + \left\{ \int_0^1 g'(x)dx \right\}^2 + \int_0^1 \{g''(x)\}^2 dx, \tag{3.6}$$

the second order Sobolev space is a RKHS with reproducing kernel

$$R(x,y) = 1 + k_1(x)k_1(y) + k_2(x)k_2(y) - k_4(|x-y|), \tag{3.7}$$

where $k_1(x) = x - \frac{1}{2}$, $k_2(x) = \frac{1}{2}\left[k_1^4(x) - \frac{1}{12}\right]$ and $k_4(x) = \frac{1}{24}\left[k_1^4(x) - \frac{1}{2}k_1^2(x) + \frac{7}{240}\right]$. Con-

trarily, when $x^{(j)}$ is a categorical variable that takes only finite distinct values, $\{1, \ldots, L\}$, we use a different reproducing kernel

$$R(s,t) = L \cdot I(s = t) - 1, \ \ s, t \in \{1, \ldots, L\}. \tag{3.8}$$

See Wahba (1990); Gu (2002) for more on reproducing kernels.

In the SS-ANOVA framework, a typical procedure is to find the $f \in \mathcal{F}$ that minimizes

$$\mathcal{L}_n(f) + \frac{\lambda}{2} \sum_{j=1}^{p} \theta_j^{-1} ||P^j f||^2, \tag{3.9}$$

where P^j is an orthogonal projection operator that projects f onto $\mathcal{F}_j$ and both λ and θ_j's are smoothing parameters. There is an over-parameterizations of λ and θ_j's but the setup is usually used for computational considerations. Denote $\tilde{f}$ be the minimizer of (3.9) and we later refer $\tilde{f}$ as an initial estimate.

For different kinds of loss functions, various authors have proposed corresponding estimation procedures. For instance, Gu (1990), Wahba et al. (1995) and Lin et al. (2000) studied a GLM where $\mathcal{L}_n$ is the negative log-likelihood function and Gu (1996, 1998) studied the bivariate hazard model, including the baseline hazard, where $\mathcal{L}_n$ is the full likelihood function.

In practice, there are other methods that can be adopted to solve an initial estimate $\tilde{f}$, but we prefer the SS-ANOVA model for several reasons. Apart from the close proximity of COSSO and SS-ANOVA, the functional space considered in the SS-ANOVA model is a RKHS, which is infinite dimensional. Moreover, SS-ANOVA model provides a flexible functional decomposition which allows us to quantify the relative importance of each function using L_2-norm. Storlie et al. (2011) studied using the inverse functional L_2-norm as an adaptive weight in the COSSO penalty and derived desirable asymptotic results in least squares context. Finally, the SS-ANOVA model enjoys desirable theoretical and computational properties. Asymptotic results for various SS-ANOVA models can be found in Gu (2002) and references therein. Several efficient tuning and estimation algorithms have been proposed and off-the-shelf computing codes are readily available.

3.2.4 Least Squares Approximation Estimator

In the approximation step, we proceed to solve (**??**) and use the adaptive COSSO penalty for $J(f)$. Thus, the NPLSA estimator is the minimizer of an adaptive COSSO problem with the initial estimate $\tilde{f}_i$ as the pseudo response. More specifically, we solve the following objective function in the approximation step to derive a final estimate

$$\min_{\boldsymbol{f}} \frac{1}{n}\|\boldsymbol{V}^{-1/2}(\boldsymbol{f}-\tilde{\boldsymbol{f}})\|^2 + \lambda\sum_{j=1}^{p} w_j||P^j f||, \tag{3.10}$$

where w_j's are known weights, which are usually the inverse functional L_2-norms. Like equation (3.1), the penalty functional in (3.10) is still required. However, the COSSO penalty not only controls the smoothness of the function, but also possesses the sparsity property. We denote $\hat{\boldsymbol{f}}$ as the minimizer of (3.10).

The existence of the minimizer of (3.10) is guaranteed due to the convexity of the objective function. Denote $R_{\mathcal{F}_j}$ as the reproducing kernel of $\mathcal{F}_j$ corresponding to the decomposition in (3.5), Storlie et al. (2011) showed the minimizer of (3.10) has a finite form

$$f(\boldsymbol{x}) = b + \sum_{i=1}^{n} c_i R_{\theta,w}(\boldsymbol{x}_i, \boldsymbol{x}), \tag{3.11}$$

where $R_{\theta,w} = \sum_{j=1}^{p}\theta_j w_j^{-2} R_{\mathcal{F}_j}$ and and $\theta_j > 0,\ j = 1,\ldots,p$, are non-negative slack variables. The slack variables θ_j's play an important role in recovering the sparse structure of f since $\theta_j = 0$ if and only if $\|P^j f\| = 0$ (Lin and Zhang, 2006). Let $\boldsymbol{R}_j$ be an $n\times n$ matrix containing elements $\{R_{\mathcal{F}_j}(x_i^{(j)}, x_{i'}^{(j)})\}_{i,i'=1}^{n}$ and $\mathbf{1}_n$ be a column vector of n ones, then we write the minimizer evaluated at the design points as $\boldsymbol{f} = b\mathbf{1}_n + (\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j)\boldsymbol{c}$. Thus, the optimization problem in (3.10) has an equivalent formulation

$$\begin{aligned}\min_{b,\boldsymbol{c},\boldsymbol{\theta}} \frac{1}{n}\left(\tilde{\boldsymbol{f}} - b\mathbf{1}_n - \sum_{j=1}^{p}\frac{\theta_j}{w_j^2}\boldsymbol{R}_j\boldsymbol{c}\right)^T \boldsymbol{V}^{-1}\left(\tilde{\boldsymbol{f}} - b\mathbf{1}_n - \sum_{j=1}^{p}\frac{\theta_j}{w_j^2}\boldsymbol{R}_j\boldsymbol{c}\right) + \lambda_0\boldsymbol{c}^T\sum_{j=1}^{p}\frac{\theta_j}{w_j^2}\boldsymbol{R}_j\boldsymbol{c} \\ \text{s.t. } \sum_{j=1}^{p}\theta_j \le M,\ \theta_j \ge 0,\ \forall j,\end{aligned} \tag{3.12}$$

where λ_0 and M are both smoothing parameters, see Storlie et al. (2011) for a detailed derivation.

An iterative algorithm that involves solving a smoothing spline problem and a non-

negative garrote problem is used in Lin and Zhang (2006) and Storlie et al. (2011) to minimize the objective function in (3.12).

3.2.5 Parameter Tuning

In penalized regression methods, information criteria, such as BIC and SIC, and cross-validation (CV) or its variant, generalized CV, are widely used in literature to select regularization parameters. A general consensus is that CV usually gives better prediction but pays the price of over-selection; whereas the information criteria could better identify the correct model. Leng et al. (2006) remarked that prediction error-based tuning procedure, like CV, will not be able to identify the correct model consistently. To fully explore the performance of NPLSA, we adopt both information criteria and CV in the following numerical study. In terms of CV, we either compute the log-likelihood or some loss function depending on the model of interest.

In order to use information criteria, an informative assessment of model complexity is required. To assess the complexity of a nonparametric procedure, a commonly-used quantity is the trace of the smoother matrix if the nonparametric procedure is a linear smoother, which is the case for many SS-ANOVA models. For instance, the degrees of freedom in a SS-ANOVA GLM model can be defined as the trace of a pseudo smoother matrix (Gu, 2002; Wang, 2011). Similarly, in the approximation step, since the COSSO algorithm requires solving a smoothing spline problem, Lin and Zhang (2006) argued the degrees of freedom of the COSSO model can be defined as the trace of the smoother matrix in the smoothing spline model.

However, the degrees of freedom derived from an individual step could be misleading considering the NPLSA method practically "smoothes" the data twice. Hence the degrees of freedom derived from individual step will over-estimate the overall one. To elaborate the mechanism how the degrees of freedom in the individual step over-estimates the overall degrees of freedom, we use a GLM as an illustrative example.

Let $y_i, i = 1, \ldots, n$, be independent observations from an exponential family (with dispersion parameter equal 1) whose density has a generic form

$$g(y_i, f_i) = \exp\{y_i f_i - b(f_i) + c(y_i)\}. \tag{3.13}$$

Lin et al. (2000) argued that a generalized degrees of freedom for an estimator $\hat{f}$ can be

defined as

$$GDF(\hat{f}) = \sum_{i=1}^{n} \text{cov}(y_i, \hat{f}_i) = \sum_{i=1}^{n} E(y_i - b'(f_i))\hat{f}_i = \sum_{i=1}^{n} b''(f_i)\frac{\partial E(\hat{f}_i)}{\partial b'(f_i)}, \tag{3.14}$$

where $E(\cdot)$ denotes the conditional expectation of y given $\boldsymbol{x}$.

Since the final estimate $\hat{f}$ in our NPLSA procedure requires an initial estimate $\tilde{f}$, we can decompose the last equality in (3.14) into

$$\sum_{i=1}^{n} \underbrace{b''(f_i)\frac{\partial E(\tilde{f}_i)}{\partial b'(f_i)}}_{Initial\ \ Step} \quad \underbrace{\frac{\partial E(\hat{f}_i)}{\partial E(\tilde{f}_i)}}_{Approximation\ \ Step}. \tag{3.15}$$

To give a graphical illustration how the degrees of freedom is decomposed, we conduct a Monte Carlo experiment to numerically evaluate the components in (3.14) and (3.15) using a toy example. We generate 10 independent predictors from $U(0,1)$ and then take $n = 200$ binary responses from a Bernoulli model using the logit function $\text{logit}(P(y = 1|\boldsymbol{x})) = \pi \sin(2\pi x^{(1)}) + \exp(x^{(2)}) - 2$. We repeated this for $B = 100$ times and hence we can estimate the required quantities in (3.14) and (3.15). In Figure 3.1, the overall degrees of freedom contribution from each observation, assessed by $\text{cov}(y_i, \hat{f}_i)$, are shown in black crosses, the initial degrees of freedom, assessed by $\text{cov}(y_i, \hat{f}_i)$, are shown in red circles and the degrees of freedom in the approximation step, assessed by $\frac{\partial E(\hat{f}_i)}{\partial E(\tilde{f}_i)}$, are shown in blue triangles. The overall degrees of freedom estimate is 8.2, whereas the SS-ANOVA model has a degrees of freedom 13.7, which is consistent with our previous claim that individual step will over-estimate the overall degrees of freedom.

After realizing the degrees of freedom in the individual step is not a good estimate of the overall one, the next question is how to better assess the complexity of our NPLSA procedure. In different kinds of regression models, corresponding generalized degrees of freedom ideas have been proposed. For instance, in a GLM framework, equation (3.14) is one option. However, equation (3.14) can not be implemented directly. For practical use, Lin et al. (2000) introduced a *rand*GACV tuning procedure to choose smoothing parameters. In addition to serving as a tuning procedure, this method also allows numerically evaluating the model degrees of freedom. The *rand*GACV is defined as

$$\textit{rand}\text{GACV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i\hat{f}_i + b(\hat{f}_i)] + \frac{1}{n}\sum_{i=1}^{n} y_i(y_i - b'(\hat{f}_i))\frac{\boldsymbol{\varepsilon}^T(\hat{\boldsymbol{f}}^{(Y+\varepsilon)} - \hat{\boldsymbol{f}})}{\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^T\hat{\boldsymbol{W}}(\hat{\boldsymbol{f}}^{(Y+\varepsilon)} - \hat{\boldsymbol{f}})}, \tag{3.16}$$

where $\hat{\boldsymbol{f}} = (\hat{f}_1, \ldots, \hat{f}_n)^T$, $\hat{\boldsymbol{W}} = \text{diag}(b''(\hat{f}_1), \ldots, b''(\hat{f}_n))$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$ are random numbers generated from a underlying distribution and $\hat{\boldsymbol{f}}^{(Y+\varepsilon)}$ is the vector of estimated function based on a perturbed response vector $y_i + \varepsilon_i,\ i = 1, \ldots, n$. The generalized degrees of freedom is defined to be n times the second term in (3.16).

In practice, we will generate $\boldsymbol{\varepsilon}$ and evaluate the ratio at the last term of equation (3.16) multiple times and use their average to produce a more stable estimate. Originally, Lin et al. (2000) suggested generating $\boldsymbol{\varepsilon}$ from $N(0, \tau^2)$ independently with tiny τ^2. But in our experimental study, the ratio was highly unstable regardless the value of τ^2. We consider a more stable alternative by drawing $\boldsymbol{\varepsilon}$ from $\{-\tau, \tau\}^n$ with probability $P(\varepsilon_i = \tau) = 0.5,\ \forall i$ and set $\tau = 0.25$ as recommended in Zhang et al. (2004).

Beyond the GLM, the degrees of freedom in nonparametric quantile regression model have also been studied by various authors (Koenker, 2005; Yuan, 2006; Li et al., 2007) and it has a simple formula to compute, the number of zero residual. However, the idea is more appropriate in the initial step. The function estimate at the final approximation step will no longer interpolate the observed response. Given that the notion of number of zero residual as a degrees of freedom estimate is originated from a more generic quantity, the divergence formula, we aim to directly estimate the divergence and use it as an effective degrees of freedom of our NPLSA procedure. The divergence formula is given by

$$\text{div}(\hat{f}) = \sum_{i=1}^{n} \frac{\partial \hat{f}_i}{\partial y_i}. \tag{3.17}$$

Similar to the generalized degrees of freedom for GLM defined in equation (3.14), the derivative in (3.17) needs to be evaluated numerically. We later consider a bootstrap method similar to Ye (1998) to estimate the derivative.

3.3 Numerical Study

3.3.1 Preliminaries

In this section, we present a series of numerical studies to demonstrate the performance of NPLSA using simulated and real data.

In simulation study, we assess the performance of NPLSA through model selection and model fidelity. We first denote $\hat{\mathcal{M}} = \{j : \hat{\theta}_j \neq 0\}$ and $\mathcal{M}_0 = \{j : ||P^j f_0|| \neq 0\}$ as the

selected and true model components, respectively, and $|\mathcal{M}|$ as the size of the set. Then we compute four criteria, correctly identify the true model, $I(\hat{\mathcal{M}} = \mathcal{M}_0)$, Type I error, $\frac{|\hat{\mathcal{M}} \cap \mathcal{M}_0^c|}{p - |\mathcal{M}_0|}$, power, $\frac{|\hat{\mathcal{M}} \cap \mathcal{M}_0|}{|\mathcal{M}_0|}$, and model size, $|\hat{\mathcal{M}}|$, to evaluate model selection. Furthermore, we use integrated squared error (ISE), $E[f_0(\boldsymbol{X}) - \hat{f}(\boldsymbol{X})]^2$, to assess function estimation. In addition to ISE, we will also incorporate appropriate criteria to assess model fitting for different regression models.

For real data application, we first compare the solution path generated by NPLSA and that by direct sparse estimator. Once we choose an optimal smoothing parameter along the path, we further compare the corresponding estimated functional profiles.

For both simulated and real data, as an exploratory experiment, we let $\boldsymbol{V} = \boldsymbol{I}$ in equation (3.12) to simplify the computation although the pointwise variance is a standard output in some off-the-shelf `R` functions. For instance, `gssanova` (GLM) and `sscox` (Cox's model).

3.3.2 Simulation Examples

Example 1: (Median Regression). In this example, we use the following functions as building blocks: $g_1(t) = t$; $g_2(t) = (2t-1)^2$; $g_3(t) = \frac{\sin(2\pi t)}{2 - \sin(2\pi t)}$ and $g_4(t) = 0.1\sin(2\pi t) + 0.2\cos(2\pi t) + 0.3\sin^2(2\pi t) + 0.4\cos^3(2\pi t) + 0.5\sin^3(2\pi t)$. We generate the response from a location-scale family

$$y_i = 5g_1(x_i^{(1)}) + 2g_2(x_i^{(2)}) + 4g_3(x_i^{(7)}) + 6g_4(x_i^{(10)}) + \sigma\varepsilon_i, \tag{3.18}$$

where $\sigma = 1$, and ε_i are independently drawn from a normal mixture $0.8\mathcal{N}(0,3) + 0.2\mathcal{N}(0,25)$. The number of predictors is 10 and each one of them is marginally $U(0,1)$ with autoregressive pairwise correlation $\text{cor}(x^{(j)}, x^{(k)}) = \rho^{|j-k|}$, $\forall j \neq k$, $\rho = \{0, 0.5\}$. We use the kernel quantile regression (KQR) (Li et al., 2007) with an additive kernel to solve an initial function estimate. In this example, The smoothing parameters in the adaptive COSSO are tuned by 5-fold CV and SIC, where SIC is defined as $\log\left(\frac{\sum_{i=1}^n \rho_\tau(y_i - \hat{f}_i)}{n}\right) + \frac{\log n}{2n} df$ and the df is derived from the divergence formula in (3.17) using bootstrap. In this example, we vary the sample size from $n = 200$ to $n = 300$.

In addition to use ISE as a model assessment, we incorporate an additional criterion for model assessment: expected check error (ECE), $E\rho_\tau(Y - \hat{f}(\boldsymbol{x}))$, where the above

expectations, including ISE, are evaluated by a Monte Carlo integration with 5,000 points generated from the same distribution as the training data.

For comparison purpose, a direct sparse estimator, a COSSO-QR model, and an Oracle estimator, a KQR model fitted with the relevant predictors, are also used to estimate the conditional median function whose smoothing parameters are tuned by 5-fold CV.

Example 2: (Logistic Regression). In this example, the dichotomous response is taken from a Bernoulli distribution using the logit function

$$\text{logit}\left(P(y_i = 1|\boldsymbol{x}_i)\right) = 2x_i^{(1)} + \pi \sin(\pi x_i^{(2)}) + 3\left(x_i^{(7)}\right)^5 + \frac{e^{x_i^{(10)}}}{e-1} - 5. \tag{3.19}$$

The predictors $x^{(j)}, j = 1, \ldots, 10$, are generated in the same fashion as in Example 1. We use the `gss` package in `R` to solve an initial function estimate $\tilde{f}$ and then apply adaptive COSSO to produce the final approximated solution. The smoothing parameters in the adaptive COSSO are tuned by 5-fold CV, BIC and *rand*GACV, whereas the degrees of freedom used in BIC is estimated by *rand*GACV. The *rand*GACV criterion has been introduced in (3.16) and the BIC is $-2\log-$likelihood$+$ $\log(n)df$. In this example, we vary the sample from $n = 250$ to $n = 350$.

In the Logistic regression model, we also include empirical misclassification rate (EMR) to assess classification power. Similarly, the above expectations, including ISE, are evaluated by a Monte Carlo integration with 5,000 points

We also include a direct sparse estimator, a COSSO-GLM model (Zhang and Lin, 2006) and an Oracle estimator, a SS-ANOVA model fitted with the relevant predictors, as references to compare with. The smoothing parameters in the direct sparse estimator and Oracle estimator are again tuned by 5-fold CV.

3.3.3 Computational Cost

Before presenting the result for two simulation examples, we first demonstrate the advantage of NPLSA in terms of computational cost. To fairly compare the computational cost, we compare the elapsed CPU time of NPLSA to that of adaptive direct sparse estimator since both methods depend on an initial estimate. We fix the smoothing parameters in

the second step of NPLSA and the direct sparse estimator at one particular value without carrying out the tuning procedure.

We run both simulation examples 200 times using different sample size and number of features combinations. All computations are done on a desktop PC with an Intel Core i7-2600K CPU and 12GB of memory. The average computation times are summarized in Table 3.1. In the median regression model, NPLSA only takes 75-85% of the computation time of the direct sparse estimator; whereas the percentage further drops to 60-70% in the Logistic regression example. Although the computational gain in the median regression example does not seem a lot, it is noteworthy that most of the time is spent on solving an initial estimate.

The save in computation time can be better elucidated by comparing the algorithms between different methods. The NPLSA depends on a COSSO algorithm which involves inverting an $n \times n$ matrix and a quadratic programming problem with p unknown parameters. COSSO-GLM requires iteratively solving COSSO problem until a certain convergence criterion is met. Finally, COSSO-QR involves a quadratic programming problem with n unknown parameters and a linear programming problem with $n + p$ unknown parameters. Hence, both COSSO-GLM and COSSO-QR are more computationally intensive than COSSO. In light of the simpler algorithm, the advantage of NPLSA becomes transparent in terms of computation.

3.3.4 Simulation Results

The result for median regression, summarized in Table 3.2 and 3.3, indicate the encouraging performance of NPLSA. When predictors are independent, the NPLSA clearly outperforms the direct sparse estimator in terms of both selection and estimation. In the correlated predictors case, NPLSA does not identify the correct model as frequently as independent case but its prediction accuracy is still close to the direct sparse estimator, most of the summary statistics are within reasonable error margins.

The simulation result for Logistic regression, shown in Table 3.4 and 3.5, again confirms the power of NPLSA. Overall, two information criteria, BIC and *rand*GACV, give similar performance and both of them identify the correct model more frequently and tend to select a smaller model size than 5-fold CV. Probably due to the tuning procedure, the direct sparse estimator commits one false positive in average and the over-selection becomes slightly worse in the dependent features case. When predictors are correlated, it

has limited influence on the performance. The selected model sizes are about the same, but the power is slightly worse. In terms of ISE, the NPLSA is still about 70% as efficient as the Oracle estimator.

3.3.5 Real Data Examples

To study the real data application, we apply NPLSA to two real data examples: South Africa heart disease data and ozone data. The heart disease data consists of 462 male patients in a heart-disease high-risk region of the Western Cape, South Africa. This data has been used by other authors (Park and Hastie, 2007; Hastie et al., 2009; Wang and Leng, 2007) to build a relationship between the disease status, with or without coronary heart disease, and 9 clinical outcomes. The ozone data contains 330 observations collected in Los Angeles in 1976, and we aim to model the median function of the daily ozone concentration on 8 meteorological covariates. Both data are publicly available from `R` packages, `ElemStatLearn` and `cosso`, respectively.

The evaluation of NPLSA procedure is first done by comparing its L_2-norm solution path to those produced by the direct sparse estimator. The solution paths for both data are illustrated in Figure 3.2. Shown on the left panel is the solution paths for heart disease data and on the right panel is that for ozone data. Overall, the solution path of the NPLSA is close to that of the direct sparse estimator. We also use 5-fold CV to determine the optimal smoothing parameter M along the path and the vertical line indicates the selected smoothing parameter. In terms of selection result for the heart disease data, the direct sparse estimator selects four functional components, whereas NPLSA selects a subset of them. As for the ozone data, both the direct sparse estimator and NPLSA select the same four components.

We next compare the estimated functional profiles of the selected components. For the heart disease data, we show the four components selected by the direct sparse method: tobacco, ldl, typea and age. As for the ozone data, we show the common four components: temp, invHt, press and vis. The estimated component functions are given in Figure 3.3. Once again, the functional profiles produced by NPLSA are very close to those produced by the direct estimator, suggesting that the NPLSA method provides a competitive performance in terms of both model selection and function estimation. Although the direct sparse estimator selects an extra predictor, typea, in the heart disease data it only shows a weak linear trend, and therefore may not be practically significant. Moreover,

according to Figure 3.3, all of the predictors in the heart disease data exhibit a linear effect on the logit function, indicating a linear model could have sufficed. On the contrary, the predictors in the ozone data suggest a highly nonlinear effect on the median function.

3.4 Discussion

In this study, we explore the applicability of LSA in a nonparametric context. Of several implementation and computation issues we address in this article, degrees of freedom estimation remains an open topic that still requires more work for other types of regression models. Model-free tuning procedure, like CV, can always be an alternative in the absence of an informative assessment of model complexity, but its tendency to under-smooth could result in rougher model fit and larger model size. The numerical result indicates a promising performance of NPLSA and our future work is to derive the theoretical properties of NPLSA.

This proposed NPLSA provides a unified framework to do variable selection and function estimation for various nonparametric regression models. Like the COSSO, the methods proposed by Meier et al. (2009) and Huang et al. (2010) also aim at joint estimation and selection in least squares problem. A similar NPLSA method can also be applied to extend the applications of their methods to other regression models.

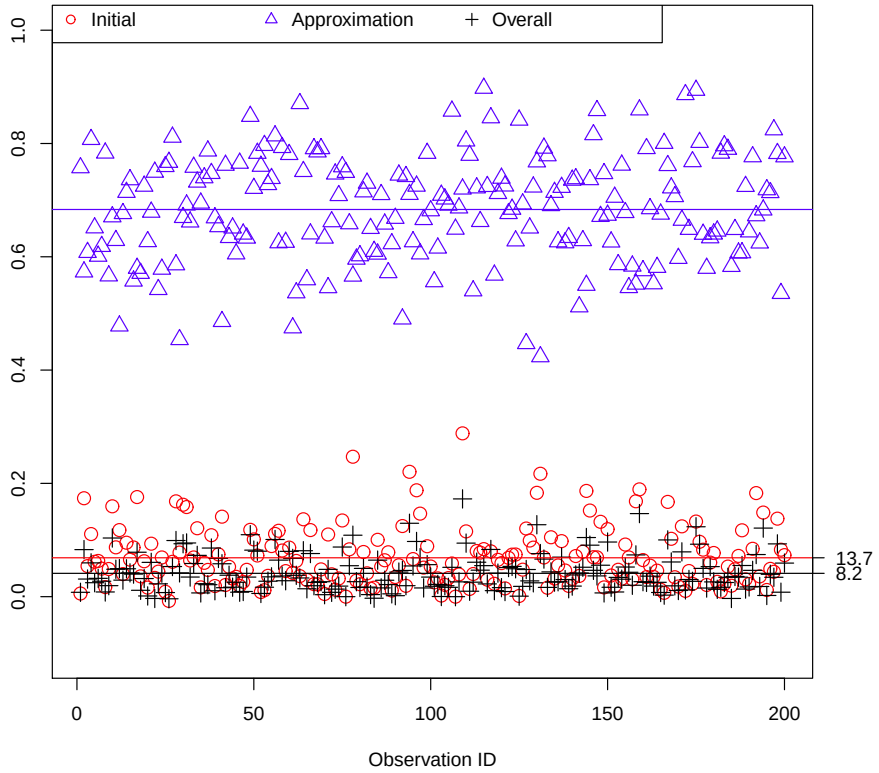

Figure 3.1: Degrees of freedom decomposition.

Table 3.1: Average elapsed CPU time (in second) for solving median regression and Logistic regression models in simulation examples 1 and 2.

	Median Regression			Logistic Regression		
(n, p)	KQR	NPLSA	ACOSSO-QR	SS-ANOVA[1]	NPLSA	ACOSSO-GLM
(200,10)	1.048	1.374	1.652	0.536	0.879	1.155
(300,10)	2.419	2.749	3.613	1.186	1.972	2.637
(200,30)	1.119	1.899	2.097	1.277	2.046	3.381
(300,30)	2.925	4.650	5.489	2.728	4.482	7.603

[1] To solve the SS-ANOVA model, we use half of the observations as "knots", the observations that used to compute the kernel matrix.

Table 3.2: Simulation result for quantile regression with independent features. The standard errors are given in the parentheses.

Method		Correct	Type I	Power	Size	ISE	ECE
				$(n, p, p_0) = (200, 10, 4)$			
COSSO-QR		0.37 (0.05)	0.11 (0.02)	0.91 (0.01)	4.31 (0.15)	1.340 (0.047)	1.073 (0.004)
NPLSA	CV	0.59 (0.05)	0.11 (0.02)	0.98 (0.01)	4.65 (0.11)	1.271 (0.045)	1.067 (0.004)
	SIC	0.65 (0.05)	0.10 (0.02)	0.98 (0.01)	4.53 (0.12)	1.238 (0.045)	1.064 (0.004)
Oracle		-	-	-	-	1.045 (0.029)	1.045 (0.003)
				$(n, p, p_0) = (300, 10, 4)$			
COSSO-QR		0.63 (0.05)	0.06 (0.02)	0.94 (0.01)	4.11 (0.11)	0.922 (0.037)	1.038 (0.004)
NPLSA	CV	0.86 (0.04)	0.03 (0.01)	1.00 (0.00)	4.16 (0.04)	0.802 (0.030)	1.028 (0.003)
	SIC	0.85 (0.04)	0.03 (0.01)	1.00 (0.00)	4.17 (0.05)	0.807 (0.030)	1.028 (0.003)
Oracle		-	-	-	-	0.690 (0.021)	1.018 (0.003)

Table 3.3: Simulation result for quantile regression with correlated features. The standard errors are given in the parentheses.

Method		Correct	Type I	Power	Size	ISE	ECE
				$(n, p, p_0) = (200, 10, 4)$			
COSSO-QR		0.22 (0.04)	0.09 (0.02)	0.85 (0.01)	3.92 (0.14)	1.301 (0.058)	1.069 (0.005)
NPLSA	CV	0.09 (0.03)	0.42 (0.03)	0.97 (0.01)	6.39 (0.14)	1.508 (0.049)	1.086 (0.004)
	SIC	0.18 (0.04)	0.02 (0.01)	0.80 (0.01)	3.34 (0.06)	1.211 (0.051)	1.061 (0.004)
Oracle		-	-	-	-	1.050 (0.042)	1.048 (0.004)
				$(n, p, p_0) = (300, 10, 4)$			
COSSO-QR		0.25 (0.04)	0.12 (0.02)	0.89 (0.01)	4.27 (0.16)	0.904 (0.032)	1.037 (0.007)
NPLSA	CV	0.10 (0.03)	0.40 (0.03)	0.99 (0.01)	6.35 (0.15)	1.042 (0.035)	1.049 (0.007)
	SIC	0.12 (0.03)	0.02 (0.01)	0.79 (0.01)	3.26 (0.05)	0.867 (0.035)	1.033 (0.007)
Oracle		-	-	-	-	0.743 (0.022)	1.022 (0.005)

Table 3.4: Simulation result for Logistic regression with independent features. The standard errors are given in the parentheses.

Method		Correct	Type I	Power	Size	ISE	EMR
				$(n, p, p_0) = (250, 10, 4)$			
COSSO-GLM		0.18 (0.04)	0.27 (0.02)	0.95 (0.01)	5.40 (0.13)	0.397 (0.019)	0.292 (0.001)
NPLSA	CV	0.16 (0.04)	0.32 (0.02)	0.96 (0.01)	5.74 (0.16)	0.388 (0.015)	0.292 (0.001)
	BIC	0.25 (0.04)	0.07 (0.01)	0.85 (0.01)	3.83 (0.09)	0.340 (0.014)	0.289 (0.001)
	*rand*GACV	0.21 (0.04)	0.12 (0.02)	0.88 (0.01)	4.19 (0.11)	0.351 (0.014)	0.289 (0.001)
Oracle		-	-	-	-	0.249 (0.011)	0.283 (0.001)
				$(n, p, p_0) = (350, 10, 4)$			
COSSO-GLM		0.28 (0.05)	0.22 (0.02)	0.94 (0.01)	5.06 (0.13)	0.293 (0.013)	0.285 (0.001)
NPLSA	CV	0.24 (0.04)	0.22 (0.02)	0.94 (0.01)	5.05 (0.15)	0.281 (0.011)	0.284 (0.001)
	BIC	0.30 (0.05)	0.05 (0.01)	0.87 (0.01)	3.74 (0.09)	0.262 (0.011)	0.284 (0.001)
	*rand*GACV	0.40 (0.05)	0.07 (0.01)	0.91 (0.01)	4.05 (0.10)	0.263 (0.012)	0.284 (0.001)
Oracle		-	-	-	-	0.202 (0.009)	0.280 (0.001)

Table 3.5: Simulation result for Logistic regression with correlated features. The standard errors are given in the parentheses.

Method		Correct	Type I	Power	Size	ISE	EMR
				$(n, p, p_0) = (250, 10, 4)$			
COSSO-GLM		0.06 (0.02)	0.34 (0.02)	0.91 (0.01)	5.69 (0.14)	0.296 (0.010)	0.350 (0.001)
NPLSA	CV	0.18 (0.04)	0.19 (0.02)	0.86 (0.02)	4.61 (0.10)	0.262 (0.010)	0.347 (0.001)
	BIC	0.17 (0.04)	0.19 (0.02)	0.84 (0.02)	4.47 (0.15)	0.281 (0.011)	0.348 (0.001)
	*rand*GACV	0.17 (0.04)	0.18 (0.02)	0.83 (0.02)	4.41 (0.14)	0.276 (0.011)	0.348 (0.001)
Oracle		-	-	-	-	0.184 (0.007)	0.341 (0.001)
				$(n, p, p_0) = (350, 10, 4)$			
COSSO-GLM		0.11 (0.03)	0.32 (0.02)	0.91 (0.01)	5.56 (0.14)	0.210 (0.010)	0.342 (0.001)
NPLSA	CV	0.24 (0.04)	0.18 (0.02)	0.88 (0.01)	4.59 (0.10)	0.184 (0.009)	0.340 (0.001)
	BIC	0.10 (0.03)	0.14 (0.02)	0.79 (0.02)	3.97 (0.15)	0.209 (0.010)	0.343 (0.001)
	*rand*GACV	0.09 (0.03)	0.15 (0.02)	0.80 (0.02)	4.10 (0.14)	0.196 (0.008)	0.342 (0.001)
Oracle		-	-	-	-	0.126 (0.006)	0.336 (0.001)

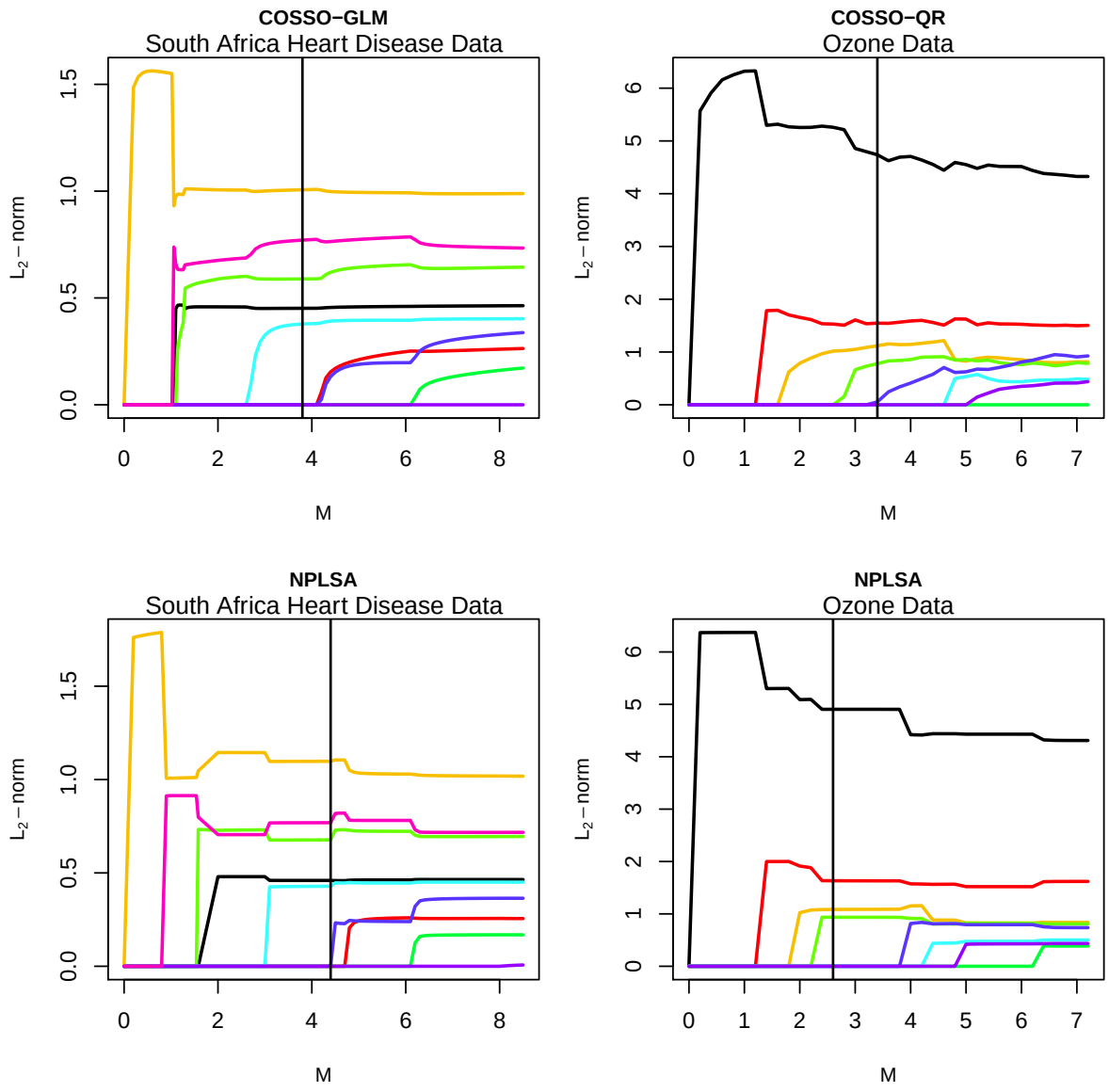

Figure 3.2: Solution paths for two real data: South Africa heart disease data (left) and ozone (right).

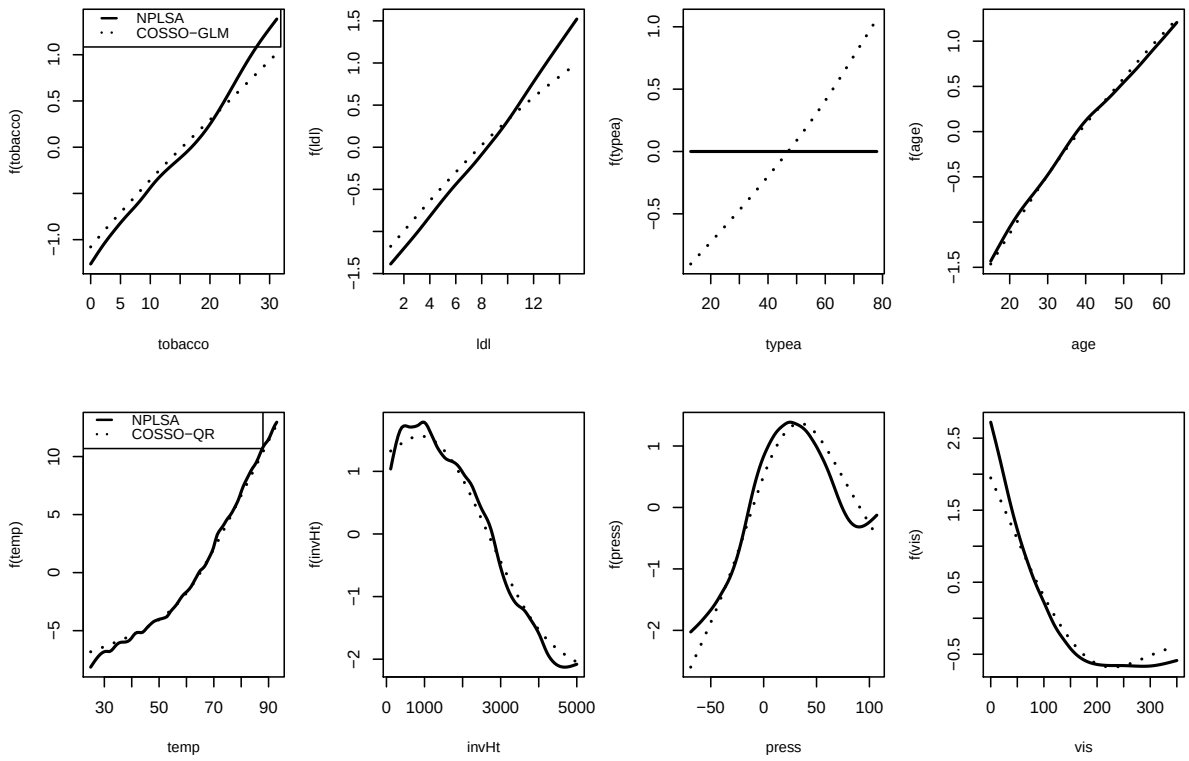

Figure 3.3: The estimated component functions for two real data: South Africa heart disease data (top) and ozone data (bottom). The solid line is for NPLSA and the dotted line is for the direct sparse estimator.

REFERENCES

P. Bloomfield and W. Steiger. *Least absolute deviations: Theory, Applications and Algorithms*. Boston: Birkhăuser-Verlag, 1983.

R. Bosch, Y. Ye, and G. Woodworth. A convergent algorithm for quantile regression with smoothing splines. *Computational Statistics and Data Analysis*, 19:613–630, 1995.

L. Breiman. Better subset selection using nonnegative garrotte. *Technometrics*, 37: 373–384, 1995.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman, J. Friedman, C. Stone, and R. A. Olshen. *Classification and Regression Tree*. Boca Raton: CRC Press, 1984.

A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models (with discussion). *Annals of Statistics*, 17:453–555, 1989.

J. Chen and Z. Chen. Extended Bayesian information criterion for model selection with large model spaces. *Biometrika*, 95:759–771, 2008.

A. P. Chiang, J. S. Beck, H. J. Yen, M. K. Tayeh, T. E. Scheetz, R. E. Swiderski, D. Nishimura, T. A. Braun, K. Y. Kim, Y. Huang, K. Elbedour, R. Carmi, D. C. Slusarski, T. L. Casavant, E. M. Stone, and V. C. Sheffield. Homozygosity mapping with SNP arrays identifies a novel gene for Bardet-Biedl syndrome (BBS10). *Proceedings of the National Academy of Sciences*, 103:6287–6292, 2006.

M. A. Clyde and H. K. H. Lee. Bagging and the Bayesian bootstrap. In T. Richardson and T. Jaakkola, editors, *Artifcial Intelligence and Statistics*, pages 169–174. New York: Springer-Verlag, 2001.

D. Donoho and V. Stodden. Breakdown point of model selection when the number of variables exceeds the number of observations. In *International Joint Conference on Neural Networks, Vancouver, Canada*, 2006.

B. Efron. The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99:619–632, 2004.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression (with discussion). *Annals of Statistics*, 32:407–499, 2004.

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

J. Fan and J. Lv. Sure independence screening for ultra-high dimensional feature space. (with discussion). *Journal of the Royal Statistical Society, Ser. B*, 70:849–911, 2008.

J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20:101–148, 2010.

J. Fan, Y. Feng, and R. Song. Nonparametric independence screening in sparse ultra-high dimensional additive model. *Journal of the American Statistical Association*, 106: 544–557, 2011.

N. Fenske, T. Kneib, and T. Hothorn. Identifying risk factors for severe childhood malnutrition by boosting additive quantile regression. *Journal of the American Statistical Association*, 106:494–510, 2011.

J. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19:1–141, 1991.

J. Friedman and B. W. Silverman. Flexible parsimonious smoothing and additive modeling (with discussion). *Technometrics*, 31:3–39, 1989.

C. Gu. Adaptive spline smoothing in non-gaussian regression models. *Journal of the American Statistical Association*, 85:801–807, 1990.

C. Gu. Penalized likelihood hazard estimation: A general procedure. *Statistica Sinica*, 6: 861–876, 1996.

C. Gu. Structural multivariate function estimation: Some automatic density and hazard estimates. *Statistica Sinica*, 8:317–335, 1998.

C. Gu. *Smoothing Spline ANOVA Models*. New York: Springler-Verlag, 2002.

T. Hastie and R. Tibshirani. *Generalized Additive Models*. London: Chapman and Hall, 1990.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data mining, Inference, and Prediction (2nd edition)*. New York: Springer-Verlag, 2009.

X. He and P. Ng. Quantile splines with several covariates. *Journal of Statistical Planning and Inference*, 75:343–352, 1999.

X. He, P. Ng, and S. Portnoy. Bivariate quantile smoothing splines. *Journal of the Royal Statistical Society, Ser. B*, 60:537–550, 1998.

J. Huang, J. L. Horowitz, and F. Wei. Variable selection in nonparametric additive models. *Annals of Statistics*, 38:2282–2313, 2010.

Z. Jin, Z. Ying, and L. J. Wei. A simple resampling method by perturbing the minimand. *Biometrika*, 88:381–390, 2001.

J. Jurečková and B. Procházka. Regression quantiles and trimmed least squares estimator in nonlinear regression model. *Journal of Nonparametric Statistics*, 3:201–222, 1994.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

J. Knaus, C. Porzelius, H. Binder, and G. Schwarzer. Easier parallel computing in R with snowfall and sfcluster. *The R Journal*, 1:47–53, 2009.

R. Koenker. *Quantile Regression*. New York: Cambridge University Press, 2005.

R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46:33–50, 1978.

R. Koenker and K. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15: 143–156, 2001.

R. Koenker, P. Ng, and S. Portnoy. Quantile smoothing splines. *Biometrika*, 81:673–680, 1994.

H. Lan, M. Chen, J. B. Flowers, B. S. Yandell, D. S. Stapleton, C. M. Mata, E. T. Mui, M. T. Flowers, K. L. Schueler, K. F. Manly, R. W. Williams, K. Kendziorski, and A. D. Attie. Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genetics*, 2:e6, 2006.

C. Leng and H. H. Zhang. Nonparametric model selection in hazard regression. *Journal of Nonparametric Statistics*, 18:417–429, 2007.

C. Leng, Y. Lin, and G. Wahba. A note on lasso and related procedures in model selection. *Statistica Sinica*, 16:1273–1284, 2006.

Y. Li and J. Zhu. L_1-norm quantile regression. *Journal of Computational and Graphical Statistics*, 17:163–185, 2008.

Y. Li, Y. Liu, and J. Zhu. Quantile regression in reproducing kernel Hilbert spaces. *Journal of the American Statistical Association*, 477:255–267, 2007.

X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *Annals of Statistics*, 28:1570–1600, 2000.

Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 34:2272–2297, 2006.

L. Meier, S. Van De Geer, and P. Bühlmann. High-dimensional additive modeling. *Annals of Statistics*, 37:3779–3821, 2009.

N. Meinshausen and P. Bühlman. Stability selection (with discussion). *Journal of the Royal Statistical Society, Ser. B*, 72:417–473, 2010.

D. Nychka, G. Gray, P. Haaland, D. Martin, and M. O'Connell. A nonparametric regression approach to syringe grading for quality improvement. *Journal of the American Statistical Association*, 90:1171–1178, 1995.

M. Y. Park and T. Hastie. L_1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Ser. B*, 69:659–677, 2007.

B. Procházka. Regression quantiles and trimmed least squares estimator in the nonlinear regression model. *Computational Statistics and Data Analysis*, 6:385–391, 1988.

D. L. Rubin. The Bayesian bootstrap. *Annals of Statistics*, 9:130–134, 1981.

T. E. Scheetz, K. Y. Kim, R. E. Swiderski, A. R. Philp, T. A. Braun, K. L. Knudtson, A. M. Dorrance, G. G. DiBona, J. Huang, T. L. Casavant, V. C. Sheffeld, and E. M. Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103:14429–14434, 2006.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:1135–1151, 1978.

T. Stamey, J. Kabalin, J. McNeal, I. Johnstone, F. Freiha, E. Redwine, and N. Yang. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate II radical prostatectomy treated patients. *Journal of Urology*, 16:1076–1083, 1989.

C. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9:1135–1151, 1981.

C. Storlie, H. Bondell, B. Reich, and H. H. Zhang. The adaptive COSSO for nonparametric surface estimation and model selection. *Statistica Sinica*, 21:679–705, 2011.

R. Tapia and J. Thompson. *Nonparametric Probability Density Estimation*. Baltimore: John Hopkins University Press, 1978.

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

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

G. Wahba. *Spline Models for Observational Data*. Philadelphia: SIAM, 1990.

G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Smoothing spline ANOVA for exponential families, with application to the Wisconsin epidemiological study of diabetic retinopathy. *Annals of Statistics*, 23:1865–1895, 1995.

H. Wang. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104:1512–1524, 2009.

H. Wang and C. Leng. Unified lasso estimation by least squares approximation. *Journal of the American Statistical Association*, 102:1039–1048, 2007.

S. Wang, B. Nan, S. Rosset, and J. Zhu. Random lasso. *Annals of Applied Statistics*, 5: 468–485, 2011.

Y. Wang. *Smoothing Splines Methods and Applications*. Boca Raton: CRC Press, 2011.

Y. Wu and Y. Liu. Variable selection in quantile regression. *Statistics Sinica*, 19:801–817, 2009.

L. Xin and M. Zhu. Stochastic stepwise ensemble for variable selection. *Journal of Computational and Graphical Statistics*, 21:275–294, 2012.

P. Yau, R. Kohn, and S. Wood. Bayesian variable selection and model averaging in high-dimensional multinomial nonparametric regression. *Journal of Computational and Graphical Statistics*, 12:23–54, 2003.

J. Ye. On measuring and correcting the effect of data mining and model selection. *Journal of the American Statistical Association*, 93:120–131, 1998.

M. Yuan. GACV for quantile smoothing splines. *Computational Statistics and Data Analysis*, 5:813–829, 2006.

H. H. Zhang and Y. Lin. Component selection and smoothing for nonparametric regression in exponential families. *Statistica Sinica*, 16:1021–1042, 2006.

H. H. Zhang, G. Wahba, Y. Lin, M. Voelker, M. Ferris, R. Klein, and B. Klein. Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association*, 99:659–672, 2004.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2567, 2006.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

H. Zou and M. Yuan. Regularized simultaneous model selection in multiple quantiles regression. *Computational Statistics and Data Analysis*, 52:5296–5304, 2008a.

H. Zou and M. Yuan. Composite quantile regression and the oracle model selection theory. *Annals of Statistics*, 36:1108–1126, 2008b.

H. Zou and H. H. Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37:1733–1751, 2009.

APPENDIX

Appendix A

Technical proofs and derivations for COSSO-QR

A.1 Existence

Proof. Denote the function to be minimized in (2.13) by

$$A(f) = n^{-1} \sum_{i=1}^{n} \rho_\tau(y_i - f(\boldsymbol{x}_i)) + \lambda J(f).$$

Without loss of generality, let $w_j = 1, \forall j$ and $\lambda = 1$. By decomposition in (2.11), for any $f \in \mathcal{F}_1$, we have $||f|| = ||\sum_{j=1}^{p} P^j f|| \leq \sum_{j=1}^{p} ||P^j f|| = J(f)$. Denote the reproducing kernel and inner product of $\mathcal{F}_1$ as $R_{\mathcal{F}_1}(\cdot,\cdot)$ and $\langle \cdot,\cdot \rangle_{\mathcal{F}_1}$. By the definition of reproducing kernel,

$$\begin{aligned} |f(\boldsymbol{x}_i)| &= |\langle f(\cdot), R_{\mathcal{F}_1}(\boldsymbol{x}_i,\cdot)\rangle_{\mathcal{F}_1}| \leq \sqrt{\langle f(\cdot), f(\cdot)\rangle}\sqrt{\langle R_{\mathcal{F}_1}(\boldsymbol{x}_i,\cdot), R_{\mathcal{F}_1}(\boldsymbol{x}_i,\cdot)\rangle_{\mathcal{F}_1}} \\ &= ||f||\sqrt{R_{\mathcal{F}_1}(\boldsymbol{x}_i,\boldsymbol{x}_i)} \leq a||f|| \leq aJ(f), \end{aligned}$$

where $a^2 = \max\limits_{i=1,\ldots,n} R_{\mathcal{F}_1}(\boldsymbol{x}_i,\boldsymbol{x}_i)$ and the first inequality holds by Cauchy-Schwarz inequality. Denote $\rho = \max\limits_{i=1,\ldots,n} |y_i|$. Consider the set

$$D(f) = \left\{ f \in \mathcal{F} : f = b + f_1, b \in \{1\}, f_1 \in \mathcal{F}_1, J(f) \leq \rho, |b| \leq (\min\{\tau, (1-\tau)\}^{-1} + a + 1)\rho \right\}.$$

Then D is a closed, convex and bounded set. By Theorem 4 of Tapia and Thompson (1978), there exists a minimizer of (2.13) in D. Denote the minimizer by $\bar{f}$. Since a constant function $f(\boldsymbol{x}) = y_{([n\tau])}$, the sample $100\tau\%$ quantile of observed y_i's, is also in D, we have $A(\bar{f}) \leq A(y_{[n\tau]}) \leq \rho$.

Conversely, if $f \notin D$, then it is either (i) $J(f) > \rho$, or (ii) $|b| > (\min\{\tau, (1-\tau)\}^{-1} + a + 1)\rho$. In case (i), we have $A(f) \geq J(f) > \rho$. Whereas in the second case, we first notice

$$\begin{aligned}\rho_\tau(y_i - b - f_1) &\geq \min\{\tau, 1-\tau\}|b - (y_i - f_1)| \geq \min\{\tau, 1-\tau\}\{|b| - |y_i| - |f_1|\} \\ &> \min\{\tau, (1-\tau)\}\{(\min\{\tau, (1-\tau)\}^{-1} + a + 1)\rho - \rho - a\rho\} = \rho,\end{aligned}$$

thus $A(f) > \rho$. Thus, for either case, we have $A(f) > A(\bar{f})$, that is $\bar{f}$ is a minimizer of (2.13).

□

A.2 Representer Theorem

Proof. Without loss of generality, let $w_j = 1,\ \forall j$. For any $f \in \mathcal{F}$, we can write $f = b + \sum_{i=1}^{p} f_j$, where $f_j \in \mathcal{F}_j$. Denote g_j as the projection of f_j onto the space spanned by $R_{\mathcal{F}_j}(\cdot,\cdot)$ and h_j as its orthogonal complement. Then $f_j = g_j + h_j$ and $||f_j||^2 = ||g_j||^2 + ||h_j||^2$. Since the reproducing kernel of $\mathcal{F}$ is $1 + \sum_{j=1}^{p} R_{\mathcal{F}_j}(\cdot,\cdot)$, by reproducing theorem, we have

$$\begin{aligned}f(\boldsymbol{x}_i) &= \left\langle 1 + \sum_{j=1}^{p} R_{\mathcal{F}_j}(x_i^{(j)},), b + \sum_{j=1}^{p} f_j \right\rangle \\ &= \left\langle 1 + \sum_{j=1}^{p} R_{\mathcal{F}_j}(x_i^{(j)},), b + \sum_{j=1}^{p} (g_j + h_j) \right\rangle \\ &= b + \left\langle 1 + \sum_{j=1}^{p} R_{\mathcal{F}_j}(x_i^{(j)},), \sum_{j=1}^{p} g_j \right\rangle + \left\langle 1 + \sum_{j=1}^{p} R_{\mathcal{F}_j}(x_i^{(j)},), \sum_{j=1}^{p} h_j \right\rangle \\ &= b + \sum_{j=1}^{p} \left\langle R_{\mathcal{F}_j}(x_i^{(j)},), g_j \right\rangle,\end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathcal{F}$.

By substituting above expression into (2.13), the objective function becomes

$$n^{-1}\sum_{i=1}^{n}\rho_\tau\left(y_i - b - \sum_{j=1}^{p}\left\langle R_{\mathcal{F}_j}(x_i^{(j)},), g_j\right\rangle\right) + \lambda\sum_{j=1}^{p}\left(||g_j||^2 + ||h_j||^2\right)^{1/2}.$$

As a result, the minimizer should be chosen such that $||h_j||^2 = 0$ and therefore completes the proof. □

A.3 Quadratic Programming Formula

Proof. To solve (2.18), we first introduce slack variables $\boldsymbol{r}_+ = (\boldsymbol{y} - b\mathbf{1}_n - (\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j)\boldsymbol{c})_+$ and $\boldsymbol{r}_- = (\boldsymbol{y} - b\mathbf{1}_n - (\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j)\boldsymbol{c})_-$, where the positive function, $(\cdot)_+$, and negative function, $(\cdot)_-$, is applied to the vector $\boldsymbol{y} - b\mathbf{1}_n - \sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j$ in an elementwise manner, then write the optimization problem in (2.18) in a matrix form

$$\min_{\boldsymbol{r}_+,\boldsymbol{r}_-,b,\boldsymbol{c}} \ \tau\mathbf{1}_n^T\boldsymbol{r}_+ + (1-\tau)\mathbf{1}_n^T\boldsymbol{r}_- + n\lambda_0\boldsymbol{c}^T\left(\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j\right)\boldsymbol{c},$$

subject to the constraints

$$\boldsymbol{r}_+ \geq 0,\ \boldsymbol{r}_- \geq 0,\ b\mathbf{1}_n + \left(\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j\right)\boldsymbol{c} + \boldsymbol{r}_+ - \boldsymbol{r}_- - \boldsymbol{y} = \mathbf{0}.$$

Then the foregoing setting gives the Lagrange primal function,

$$\begin{aligned}\mathcal{L} \ = \ & \tau\mathbf{1}_n^T\boldsymbol{r}_+ + (1-\tau)\mathbf{1}_n^T\boldsymbol{r}_- + n\lambda_0\boldsymbol{c}^T\left(\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j\right)\boldsymbol{c} + \\ & \boldsymbol{\lambda}_1^T\left[b\mathbf{1}_n + \left(\sum_{j=1}^{p}\theta_j w_j^{-2}\boldsymbol{R}_j\right)\boldsymbol{c} + \boldsymbol{r}_+ - \boldsymbol{r}_- - \boldsymbol{y}\right] - \boldsymbol{\lambda}_2^T\boldsymbol{r}_+ - \boldsymbol{\lambda}_3^T\boldsymbol{r}_-,\end{aligned}$$

where $\boldsymbol{\lambda}_1 \in \mathbb{R}^n, \boldsymbol{\lambda}_2 \geq \mathbf{0}, \boldsymbol{\lambda}_3 \geq \mathbf{0}$ are Lagrange multipliers. By differentiating $\mathcal{L}$ with respect to $\boldsymbol{r}^+$, $\boldsymbol{r}^-$, b and $\boldsymbol{c}$, we arrive at

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{r}_+} &: \tau \mathbf{1}_n + \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 = \mathbf{0} \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{r}_-} &: (1-\tau)\mathbf{1}_n - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_3 = \mathbf{0} \\
\frac{\partial \mathcal{L}}{\partial b} &: \boldsymbol{\lambda}_1^T \mathbf{1}_n = 0 \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{c}} &: 2n\lambda_0 \left(\sum_{j=1}^{p} \theta_j w_j^{-2} \boldsymbol{R}_j \right) \boldsymbol{c} + \left(\sum_{j=1}^{p} \theta_j w_j^{-2} \boldsymbol{R}_j \right) \boldsymbol{\lambda}_1 = \mathbf{0}.
\end{aligned}$$

By substituting these conditions into the Lagrange primal function, the dual problem is given by

$$\min_{\boldsymbol{c}} -\boldsymbol{y}^T \boldsymbol{c} + \frac{1}{2} \boldsymbol{c}^T \left(\sum_{j=1}^{p} \theta_j w_j^{-2} \boldsymbol{R}_j \right) \boldsymbol{c},$$

subject to the constraints

$$\mathbf{1}_n^T \boldsymbol{c} = 0, \ \frac{\tau - 1}{2n\lambda_0} \mathbf{1}_n \leq \boldsymbol{c} \leq \frac{\tau}{2n\lambda_0} \mathbf{1}_n.$$

□

A.4 Linear Programming Formula

Proof. To solve (2.19), we take similar route as solving (2.18) by introducing slack variables $\boldsymbol{e} = |\boldsymbol{y}^* - \boldsymbol{G\theta}|, \boldsymbol{e}_+ = (\boldsymbol{y}^* - \boldsymbol{G\theta})_+$ and $\boldsymbol{e}_- = (\boldsymbol{y}^* - \boldsymbol{G\theta})_-$ and re-write the objective function into a matrix form

$$\begin{aligned}
\tau \mathbf{1}_n^T \boldsymbol{e}_+ + (1-\tau)\mathbf{1}_n^T \boldsymbol{e}_- + n\lambda_0 \boldsymbol{c}^T \boldsymbol{G\theta} &= \mathbf{1}_n^T \boldsymbol{e}_- + \tau \mathbf{1}_n^T (\boldsymbol{e}_+ - \boldsymbol{e}_-) + n\lambda_0 \boldsymbol{c}^T \boldsymbol{G\theta} \\
&= \frac{1}{2} \mathbf{1}_n^T \boldsymbol{e} - \frac{1}{2} \mathbf{1}_n^T (\boldsymbol{e}_+ - \boldsymbol{e}_-) + \tau \mathbf{1}_n^T (\boldsymbol{y}^* - \boldsymbol{G\theta}) + n\lambda_0 \boldsymbol{c}^T \boldsymbol{G\theta} \\
&= \frac{1}{2} \mathbf{1}_n^T \boldsymbol{e} + \left(\tau - \frac{1}{2} \right) \mathbf{1}_n^T (\boldsymbol{y}^* - \boldsymbol{G\theta}) + n\lambda_0 \boldsymbol{c}^T \boldsymbol{G\theta}.
\end{aligned}$$

Since $\left(\tau - \frac{1}{2}\right) \mathbf{1}_n^T \boldsymbol{y}^*$ is a constant, the objective function can be simplified to

$$\min_{\boldsymbol{\theta},\boldsymbol{e}} \begin{pmatrix} n\lambda_0 \boldsymbol{c}^T \boldsymbol{G} - (\tau - 0.5)\mathbf{1}_n^T \boldsymbol{G} & \frac{1}{2}\mathbf{1}_n^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{e} \end{pmatrix},$$

subject to the constraints

$$\mathbf{1}_p^T \boldsymbol{\theta} \leq M,\ \theta_j \geq 0,\ \forall j, \quad -\boldsymbol{e} \leq \boldsymbol{y}^* - \boldsymbol{G}\boldsymbol{\theta} \leq \boldsymbol{e}.$$

□