ABSTRACT

DEMIRHAN, EREN. Variable Selection For Multivariate Smoothing Splines With Correlated Random Errors. (Under the direction of Hao Helen Zhang.)

Variable selection in multivariate nonparametric regression is an important but challenging problem, which becomes even more difficult for correlated data such as time series data, spatial data, longitudinal data, and repeated measurements. Very little work exists for nonparametric variable selection with correlated data. In the framework of smoothing spline analysis of variance (SS-ANOVA), we propose some unified approaches to simultaneously select important variables, estimate the multivariate nonparametric function, and estimate the variance components. In particular, two methods are proposed: the Correlated COSSO (Cor-COSSO), which is a generalization of the component selection and smoothing operator (COSSO; Lin and Zhang 2006) to correlated random errors, and the Adaptive Correlated COSSO, which is an improvement on Correlated COSSO. We show that, the Cor-COSSO solves the penalized weighted least squares subject to a soft-thresholding penalty on the functional components, which encourages sparse estimation while taking the data covariance structure into account at the same time. The Adaptive Correlated COSSO introduces a set of adaptive weights in the penalty term, which results in different scales of penalization on different components. We study the existence of the solution to the proposed regularization problems, and show that the minimizer possesses the desired finite dimensional representation property as standard smoothing splines (Kimeldorf and Wahba, 1971). One important issue in the SS-ANOVA model estimation for correlated data is the selection of smoothing parameters (Wang, 1998b;

Opsomer, Wang, and Yang, 2001). We show that both the Cor-COSSO and the Adaptive Cor-COSSO nicely handle this difficulty by estimating the smoothing parameters and variance components at the same time with the generalized maximum likelihood (GML - Wang 1998b) estimation. In addition, we develop efficient computational algorithms, which solve the proposed methods by iteratively solving a quadratic programming (QP) problem and fitting a linear mixed effects model. Therefore, the Cor-COSSO and Adaptive Cor-COSSO can be conveniently implemented by standard software packages. We demonstrate the performance of these methods through extensive simulations and real examples, and compare them with other competitive methods in various settings.

Variable Selection for Multivariate Smoothing Splines with Correlated Random Errors

by

Eren Demirhan

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

2008

APPROVED BY:

Dr. Daowen Zhang	Dr. Jason A. Osborne
Dr. Hao H. Zhang Chair of Advisory Committee	Dr. Sujit K. Ghosh

DEDICATION

To my family

BIOGRAPHY

Eren Demirhan was born in 1981 in Izmir, Turkey. He is the son of Nurcan and Haci Demirhan. He attended Cankaya High School (ACL) in Ankara, Turkey. Following his graduation from high school in 1998, he was admitted to Middle East Technical University (METU) where he received her Bachelor of Arts degree in Statistics with honors in 2002. He participated in a non-degree graduate study in Biostatistics at Ankara University School of Medicine, Department of Biostatistics. He moved to the United States in 2003 to pursue his graduate education in Statistics in Raleigh, North Carolina. He was graduated with Master of Statistics in North Carolina State University, Department of Statistics, and is pursuing for the Philosophy of Doctorate degree within the same department.

ACKNOWLEDGMENTS

I would like to express my gratefulness to Dr. Hao Helen Zhang, my advisor, for her constant guidance, support, and encouragement. I would like to thank her especially for believing in me sometimes even more than I did in myself. I would also like to thank my advisory committee members Dr. Sujit Ghosh, Dr. Daowen Zhang, Dr. Jason Osborne, and Dr. Steve H. Barr for their support and valuable suggestions.

Last but not least, I would like to thank my friends and my family for their patience and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Linear Models and Variable Selection

The general purpose in regression modelling is to explain the relationship between the response variable and some predictor variables using the limited number of observations at hand. A common representation of the regression model with additive error terms is:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, ..., n, \tag{1.1}$$

where y_i's are dependent observations, $\mathbf{x}_i = (x_i^{(1)}, \ldots, x_i^{(p)})^{\mathrm{T}}$ is a p-dimensional vector of explanatory variables, and the error terms ε_i's are zero mean random variables. The variance-covariance structure of the error terms is intentionally left undefined at this moment.

A popular approach to find an approximation to the regression function $f(\mathbf{x})$ is parametric regression modeling, where a specific formulation for $f(\mathbf{x})$ is assumed. In particular, the linear regression model $f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x^{(j)}$ assumes that the relationship between the response and the predictors is linear in parameters (β_j's). The linear regression model is extensively studied in the literature, and almost coincides with the name *regression* in introductory statistics textbooks. These models require relatively a small number of data points and they are easy to fit and interpret. Computation of these models is very fast (Drapper and Smith, 1998).

As one consequence of the information era and with the advances in technology, high-dimensional data sets become more and more common, especially in genetics, environmental and medical sciences. In these high dimensional data situations, reducing data is very important in order to improve both interpretability and prediction accuracy of the model. The regression model is mostly used to make predictions on future observations, and the analyst would like to have narrow interval estimates for these predictions. In multivariate regression settings, some of the explanatory variables might not be affective on prediction, however, these uninformative variables increase the prediction variance, and hence diminish the prediction accuracy. On the other hand, discovering which variables are beneficial in prediction itself helps scientists make better interpretations of the model. It is always desirable and more practical to evaluate a regression model containing a smaller number of variables.

There is a vast literature on variable selection in linear models. Traditional approaches to variable selection are based on penalizing the number of variables included in the model. Best subset selection, forward/backward/stepwise selection, Mallow's C_p, Akaike (AIC - Akaike 1973) and Bayesian Information criteria (BIC - Schwarz 1978) are well known examples of traditional methods (Miller, 1990). These methods either exclude some of the explanatory variables from the model, or retain the regression coefficients corresponding to those variables intact. The discrete nature of these methods can suffer from extremely variable results, such that even small changes in data may result in very different models to be selected (Tibshirani, 1996).

The instability and lack of accuracy of these models are pointed out in Breiman (1995). A class of shrinkage methods called Bridge regression (Frank and Friedman, 1993) is proposed, and followed by a set of variable selection methods using the shrinkage idea such as Nonnegative Garrote (Breiman, 1995), LASSO (Tibshirani, 1996), SCAD (Fan and Li, 2001), and Elastic Net (Zou and Hastie, 2005) and etc. These methods apply continuous shrinkage to regression coefficients. Some of the coefficients are shrunk to 0, and hence corresponding variables are excluded from the model. The promising properties of these methods - especially their good performance in prediction accuracy - are presented in corresponding articles, and they have been used widely in applied statistics.

There is a whole literature on Bayesian variable selection models. These models assign a hierarchial Bayesian mixture priors, and uses these priors to calculate the posterior probabilities of the candidate models (George and McCulloch, 1993). For models with large number of explanatory variables to start with, the calculation of the 2^p posterior probabilities might be overwhelming. Some of the methods use the Gibbs sampling algorithm, such as Stochastic Search Variable Selection (George and McCulloch, 1995, 1993; Chipman, 1996) to calculate the posterior probabilities. Bayesian model selection methods use these posterior probabilities to select the final model. Some examples for these methods can be found in George and McCulloch (1997), Geweke (1996), Casella and Moreno (2006) and Berger and Pericchi

(2001).

1.2 Nonparametric Regression Models

Although computational and interpretational advantages of linear models are appealing, the assumption on the form of the relationship between the response and explanatory variables is restrictive. In other words, the linearity assumption limits the flexibility of regression models. If the underlying regression function cannot be approximated linearly, the results derived from the linear model will not be accurate, and will be even misleading sometimes. For details on the effect of misspecifications in linear models, reader is referred to the discussion in Chapter 1 of Eubank (1988).

To overcome the possible problems arising from the restrictive assumption of the function forms in parametric models, nonparametric regression techniques can be used. These methods do not assume any specific form for $f(\mathbf{x})$. Instead, they search for the regression function that best matches the data in a much larger function space where less restrictions are imposed on this form. Caution should be given that using nonparametric regression does not mean that no assumption is made on $f(\mathbf{x})$. However, the restrictions are minimal, and generally limited with smoothness conditions. For example a cubic spline model assumes $f(\mathbf{x})$ to be smooth enough to have continuous derivatives up to second order, and the second derivative to be squared integrable.

Another important feature of the linear regression models is their global nature. The model is defined over an extensive range of the explanatory variables, which allows for extrapolation, i.e., prediction for future observations which are not included in the range of the original data set from which the model is fitted. Although this looks like an advantage at the first glance (and it is for some specific situations), the vulnerability of these models to influential observations turns the global model structure into a disadvantage. In other words, these models might become very sensitive to outliers. On the contrary, nonparametric regression methods are local methods, therefore are more robust, i.e., the outliers will have less effect in nonparametric methods compared to their linear counterparts.

Less restrictive assumptions of nonparametric regression allow the data to define the functional dependence of the response to explanatory variables. These methods search a larger function space to find the *best* estimator for $f(\mathbf{x})$. The price of this flexibility is mostly the computation time and efficiency loss if the true model is linear. Especially in multivariate regression settings, nonparametric methods face the challenges in computation time and interpretability, and many suffer from the curse of dimensionality. For a more detailed introduction to nonparametric regression methods, reader is referred to Eubank (1988) and Green and Silverman (1988).

The curse of dimensionality is a bottleneck for multivariate nonparametric regression methods. One alternative is to use the popular additive models summarized in Hastie and Tibshirani (1990). Under the arguably restrictive assumption of additive models, they proposed to approximate p-dimensional surface by a finite sum of p-univariate smoothers. The backfitting algorithm can be used to fit additive models. This method is very popular and has been widely applied in various disciplines.

Another model building tool for multivariate nonparametric regression is Multivariate Adaptive Regression Splines (MARS, Friedman 1991). The method works in a regression spline framework, and can be considered as an extension of additive models. MARS can be considered as a generalization of both binary partitioning and stepwise linear regression (see Hastie, Tibshirani, and Friedman 2001). Using so called *reflected pairs*, the method builds up a large model with forward selection techniques. After a large model is achieved, pruning is conducted via backward elimination, and least effective components (reflected pairs) in prediction are eliminated from the model. Some variables lose all their reflected pairs from the model during pruning, and therefore are excluded entirely. In other words, MARS works as a variable selection method in nonparametric setting, and it is one of the most famous methods in this area. For more information on MARS, reader is referred to Friedman (1991) and Hastie, Tibshirani, and Friedman (2001).

There are two popular approaches in the smoothing spline framework for multivariate nonparametric regression: thin plate splines and smoothing spline analysis of variance (SS-ANOVA hereafter). They are both useful for estimation of multidimensional smooth functions. Our proposal is developed in the SS-ANOVA framework, and we will explain the close relationship while presenting the algorithm. Therefore, the SS-ANOVA models will be crucial in the progress of this research, and a quick review of them will be given in the following part.

1.3 Smoothing Spline ANOVA Models

In this section we will introduce univariate smoothing spline models and their extension to multivariate smoothing splines analysis of variance models. For now we focus on the independent error structure. The models for correlated data will be incorporated in the following sections.

1.3.1 One-dimensional Smoothing Spline with Independent Data

We first consider the regression problem in (1.1) by assuming the error terms are independent and have zero mean and constant variance σ^2. We will concentrate on the smoothing spline method, which is a popular nonparametric regression model with elegant mathematical framework and theoretical properties.

Smoothing spline is a regularization method, where the model complexity is controlled by a smoothing parameter. Assume that $f(\mathbf{x})$, $\mathbf{x} \in \mathscr{T}$, lies in some prechosen function space $\mathscr{H}$, generally satisfying some smoothness conditions. For this part, we will focus on one dimensional smoothing spline problem, where the design matrix contains only one explanatory variable. Later we will extend the definition to multivariate smoothing splines (and SS-ANOVA models). For a discussion of model spaces for various index sets, see Wahba (1990) and Gu (2002).

Here is a very short introduction to reproducing kernel Hilbert spaces. The background for RKHS spaces can be found in Halmos (1957) and Aronsajn (1950). A shorter introduction to RKHS spaces and their uses in smoothing spline framework can be found in Wahba (1990) and Gu (2002). A functional in a linear space $\mathscr{L}$ is a mapping of an element in $\mathscr{L}$ to a real number in real line $\mathbb{R}$. A linear functional L in $\mathscr{L}$ satisfies $L(f+g) = Lf + Lg, L(\alpha f) = \alpha Lf$ where $f, g \in \mathscr{L}$, and $\alpha \in \mathbb{R}$. A bilinear form $J(\cdot,\cdot) : \mathscr{L} \times \mathscr{L} \to \mathbb{R}$ satisfies $J(\alpha f + \beta g, h) = \alpha J(f,h) + \beta J(g,h)$, $J(f, \alpha g + \beta h) = \alpha J(f,g) + \beta J(f,g)$ where $f, g, h \in \mathscr{L}$ and $\alpha, \beta \in \mathbb{R}$. A *linear space* (also known as a vector space) often is equipped with an *inner product*, which is a positive definite bilinear form defined as $< \cdot, \cdot >$. This inner product defines a *norm* in $\mathscr{L}$ such as $\|f\| =< f, f >^{1/2}$, which also induces a metric to measure the distance between the elements of $\mathscr{L}$ such that $D(f,g) = \|f - g\|$. This space $\mathscr{L}$ equipped with the inner product $< \cdot, \cdot >_{\mathscr{L}}$ is called an inner product space.

A Hilbert space ($\mathscr{C}$) is a complete inner product linear space. An important property of Hilbert spaces is the Reisz representation theorem, which states that for every *continuous linear functional* L in a Hilbert space $\mathscr{C}$, there exists a unique $g_L \in \mathscr{C}$ such that for any $f \in \mathscr{C}$, $Lf =< g_L, f >$. Here, g_L is called the representer of the linear functional L.

We use this machinery in order to maximize the penalized weighted least squares functional, which usually involves evaluations at data points. Let $\mathscr{H}$ be a Hilbert space of real valued functions on a domain $\mathscr{T}$, where the evaluation functional is defined as $\mathscr{L}_x(f) = f(x)$. If $\mathscr{L}_x(f)$ is bounded $\forall x \in \mathscr{T}, \forall f \in \mathscr{H}$, then the space $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS hereafter). In other words, if for each $x \in \mathscr{T}$, there exists M_x such that $|f(x)| \leq M_x \|f\|_{\mathscr{H}}, \forall f \in \mathscr{H}$ then $\mathscr{H}$ is an RKHS.

Although many results for smoothing splines can be obtained without the RKHS assumption, this assumption both saves one from proving the same theorems over and over again (Wahba, 1990), and provides a very important link between Bayesian estimation. Without loss of generality, we will use the domain $\mathscr{T}$ of explanatory variable as $[0,1]$. Any range can be easily transformed to fit in this domain, therefore, $\mathscr{T}$ is very general, and covers any continuous domain. A common example of RKHS widely used in smoothing spline framework is the m^{th}-order Sobolev space:

$$S_2^m = \{f : f^{(\nu)} \text{ is absolutely continuous}, \quad \nu = 0,1,\ldots,m-1 \quad \text{and} \int_0^1 (f^{(m)}(t))^2 dt < \infty\}.$$

When equipped with the inner product

$$< f,g >_{\mathscr{H}} = \sum_{\nu=0}^{m-1} f^{(\nu)}(0)g^{(\nu)}(0) + \int_0^1 f^{(m)}(t)g^{(m)}(t)dt, \tag{1.2}$$

the space S_2^m is an RKHS.

In general, we decompose $\mathscr{H} = S_2^m = [1] \oplus \mathscr{H}_1$, where $[1]$ contains the constant functions in $\mathscr{H}$, and $\mathscr{H}_1$ is the complement subspace of $[1]$. Define $\mathbf{y} = (y_1,\ldots,y_n)^\mathrm{T}$ and $\mathbf{f} = (f(x_1),\ldots,f(x_n))^\mathrm{T}$. The penalized least squares for estimating f is obtained by solving:

$$\min_{f \in \mathscr{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda ||P^1 f||_{\mathscr{H}}^2, \tag{1.3}$$

where $\lambda > 0$ is the smoothing parameter, P^1 is the projection operator onto $\mathscr{H}_1$, and $||P^1 f||_{\mathscr{H}}^2$ is the squared norm reflecting the roughness of f. The tuning parameter $\lambda > 0$ balances the tradeoff between the data fit and the function smoothness. For different specifications of λ, the smoothing spline fit can range from a linear regression fit to a data interpolation. Therefore, in order to acquire a good representation from the smoothing spline, an appropriate selection of the tuning parameter is crucial.

The reproducing kernel (RK) of $\mathscr{H}_1$ associated with the inner product (1.2) is given by:

$$R_1(s,t) = \sum_{\nu=0}^{m-1} \frac{s^\nu}{\nu!}\frac{t^\nu}{\nu!} + \int_0^1 \frac{(s-u)_+^{m-1}}{(m-1)!}\frac{(t-u)_+^{m-1}}{(m-1)!}du, \tag{1.4}$$

where $(x)_+ = \max\{x, 0\}$. Correspondingly, the penalty term in (1.3) becomes:

$$\|P^1 f\|_{\mathscr{H}}^2 = \sum_{\nu=0}^{m-1} [f^{(\nu)}(0)]^2 + \int_0^1 \left(f^{(m)}(t)\right)^2 dt. \tag{1.5}$$

The most popular smoothing spline model is the cubic spline with $m = 2$. It can be seen that in cubic spline the reproducing kernel $R_1(s,t)$ has a simpler expression as:

$$R_1(s,t) = st + (s \wedge t)^2 (3(s \vee t) - (s \wedge t))/6, \tag{1.6}$$

where $s \wedge t = \min(s,t)$ and $s \vee t = \max(s,t)$. The minimizer of (1.3) is known as the cubic smoothing spline; see Green and Silverman (1988), Wahba (1990) and Gu (2002) for more details.

1.3.2 Smoothing Spline ANOVA Models

Now let us move on to the problem of multidimensional function estimation, where $\mathbf{x} = (x^{(1)}, \ldots, x^{(p)})^{\mathrm{T}}$ is a p-dimensional input vector from the p-dimensional index set $\mathscr{T} = \mathscr{T}^{(1)} \times \cdots \times \mathscr{T}^{(p)}$. The goal is to estimate a multivariate function $f(\mathbf{x})$ given the data points $\{\mathbf{x}_i, y_i\}, i = 1, \cdots, n$. Smoothing spline ANOVA model (SS-ANOVA) provides a general framework for high dimensional function estimation, and has been successfully applied to many practical problems. See Wahba (1990), Wahba et al. (1995), and Gu (2002) for details.

The functional ANOVA decomposition of any p-variate function f is:

$$f(\mathbf{x}) = d + \sum_{j=1}^{p} f_j(x^{(j)}) + \sum_{j=1}^{p} \sum_{k=j+1}^{p} f_{jk}(x^{(j)}, x^{(k)}) + \cdots, \tag{1.7}$$

where d is a constant, f_j's are the main effects, f_{jk}'s are the two-way interactions, and so on. The identifiability of the terms in (1.7) is assured by side conditions through averaging operators.

The main purpose of the functional ANOVA decomposition is to decompose the multivariate surface into a sum of univariate functions, and estimate the whole surface using these univariate functions. This will facilitate the estimation computationally. We assume each component (f_j's) belongs to a univariate function space $\mathscr{H}^{(j)}$ over $\mathscr{T}^{(j)}$; and $\mathscr{H}^{(j)} = [1] \oplus \mathscr{H}_1^{(j)}$ where $[1]$ consists of the constant functions and $\mathscr{H}_1^{(j)}$ is the complement of the subspace $[1]$ for each $j = 1, \ldots, p$. Then the full tensor product space $\mathscr{H} = \otimes_{j=1}^{p} \mathscr{H}^{(j)}$ has the tensor sum decomposition:

$$\mathscr{H} = \otimes_{j=1}^{p} \mathscr{H}^{(j)} = \otimes_{j=1}^{p} ([1] \oplus \mathscr{H}_1^{(j)}) = \oplus_{\mathscr{S}} \left\{ \otimes_{j \in \mathscr{S}} \mathscr{H}_1^{(j)} \right\} = \oplus_{\mathscr{S}} \mathscr{H}_{\mathscr{S}}, \qquad (1.8)$$

where the summation is over all possible subsets $\mathscr{S} \subset \{1, \ldots, p\}$. Each term $\mathscr{H}_{\mathscr{S}}$, as a subspace of $\mathscr{H}$, is also an RKHS. For ease of interpretation, we usually truncate $\mathscr{H}$ in (1.8) by keeping only lower order terms and conduct the estimation in the subspace:

$$\mathscr{H} = [1] \oplus_{j=1}^{q} \mathscr{H}^{j}, \qquad (1.9)$$

where $\mathscr{H}^1, ..., \mathscr{H}^q$ are q orthogonal subspaces of $\mathscr{H}$. Here j is a generic index, and $\mathscr{H}^j$ has the inner product $< f_j, g_j >_{\mathscr{H}^j}$ and the reproducing kernel R_j, where f_j is the projection of f onto $\mathscr{H}^j$. According to Wahba (1990) and Gu (2002), we can define the inner product in $\mathscr{H}$ as:

$$< f, g >_{\mathscr{H}} = \sum_{j=1}^{q} \theta_j^{-1} < f_j, g_j >_{\mathscr{H}^j}, \qquad (1.10)$$

where θ_j's are non-negative tuning parameters. The reproducing kernel of $\mathscr{H}$ in (1.9) is the weighted sum of kernels in individual spaces. Therefore, the reproducing kernel associated with (1.9) can be written as:

$$R_\theta(\cdot, \cdot) = \sum_{j=1}^{q} \theta_j R_j(\cdot, \cdot), \qquad (1.11)$$

since

$$< R_\theta(\mathbf{x},\cdot), f(\cdot) >_{\mathscr{H}} = \sum_{j=1}^{q} \theta_j^{-1} < \theta_j R_j(\mathbf{x},\cdot), f_j >_{\mathscr{H}^j} = \sum_{j=1}^{q} f_j(\mathbf{x}) = f(\mathbf{x}).$$

At this moment, we assume uncorrelated error terms, i.e., $\varepsilon \sim N(0,\sigma^2 I)$. The SS-ANOVA model for data with independent errors aims to find $f \in \mathscr{H}$ which minimizes the following penalized least squares equation:

$$\min_{f\in\mathscr{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{q} \theta_j^{-1}\|P^j f\|^2. \tag{1.12}$$

Here P^j is the projection operator onto $\mathscr{H}^j$, $\sum_{j=1}^{q}\theta_j^{-1}\|P^j f\|^2$ is the squared norm reflecting the roughness of f, and $\lambda, \theta_1, \dots \theta_q > 0$ are smoothing parameters. The main purpose of this dissertation research is to work with correlated error terms, therefore, we will relax this assumption in following sections.

1.3.3 Computation of SS-ANOVA Models

The computation for SS-ANOVA model with fixed smoothing parameters λ and θ's is very similar to univariate smoothing spline computation. We will only cover the multivariate case in this section.

Let $\mathbf{1}$ be the vector of ones with length n. Given the sample points $\mathbf{x}_i$, $i = 1, \dots, n$, with some abuse of notation define the $n \times n$ Gram matrix R_θ with entries $R_{\theta,ii'} = R_\theta(\mathbf{x}_i, \mathbf{x}_{i'})$, where $R_\theta(\cdot,\cdot)$ is defined in (1.11). The multidimensional space $\mathscr{H}$ is an RKHS, and we can calculate the reproducing kernel for this space using the univariate RK's.

The representer theorem of Kimeldorf and Wahba (1971) proves that the SS-ANOVA estimate lies in a finite dimensional space, and can be written in the form:

$$f(\mathbf{x}) = d + \sum_{i=1}^{n} c_i R_\theta(\mathbf{x}, \mathbf{x}_i), \tag{1.13}$$

where d and $\mathbf{c} = (c_1, \ldots, c_n)^{\mathrm{T}}$ can be estimated by minimizing

$$(\mathbf{y} - \mathbf{1}d - R_\theta \mathbf{c})^{\mathrm{T}}(\mathbf{y} - \mathbf{1}d - R_\theta \mathbf{c}) + n\lambda \mathbf{c}^{\mathrm{T}} R_\theta \mathbf{c}. \tag{1.14}$$

It can be shown that the SS-ANOVA estimate $\hat{\mathbf{f}} = \mathbf{1}\hat{d} + R_\theta \hat{\mathbf{c}}$ exists and is unique (Wahba 1990, Gu 2002). Using some algebra we can show that the solution to (1.14) satisfies:

$$\begin{aligned} (R_\theta + n\lambda I)\,\hat{\mathbf{c}} + \mathbf{1}\hat{d} &= \mathbf{y}, \\ \mathbf{1}^{\mathrm{T}}\hat{\mathbf{c}} &= \mathbf{0}. \end{aligned} \tag{1.15}$$

To facilitate the computation of $\hat{\mathbf{c}}$ and $\hat{d}$ in (1.15), one can use the QR decomposition of $\mathbf{1}$

$$\mathbf{1} = (Q_1 \quad Q_2) \begin{pmatrix} S \\ 0 \end{pmatrix},$$

where Q_1 is $n \times 1$ and Q_2 is $n \times (n-1)$, $Q = [Q_1 \;\; Q_2]$ is orthogonal and S is upper triangular, with $\mathbf{1}^{\mathrm{T}}Q_2 = 0$. Using this new formulation, the solution can be rephrased as:

$$\hat{\mathbf{c}} = Q_2\,[Q_2^{\mathrm{T}}(R_\theta + n\lambda I)Q_2]^{-1}\,Q_2^{\mathrm{T}}\mathbf{y},$$
$$\hat{d} = S^{-1}\,[Q_1^{\mathrm{T}}\mathbf{y} - Q_1^{\mathrm{T}}R_\theta\hat{\mathbf{c}}]\,.$$

The SS-ANOVA estimate $\hat{\mathbf{f}}$ is linear in $\mathbf{y}$, i.e., $\hat{\mathbf{f}} = \mathbf{A}\mathbf{y}$, where $\mathbf{A}$ is known as the *hat* or the *influence* matrix. Here $\mathbf{A}$ is symmetric and can be written as:

$$\mathbf{A} = I - n\lambda Q_2\,[Q_2^{\mathrm{T}}R_\theta Q_2 + n\lambda I]^{-1}\,Q_2^{\mathrm{T}}.$$

As mentioned above, the solution to univariate smoothing spline is very similar to SS-ANOVA. The only difference is the function space ($\mathscr{H}$), which will be a univariate RKHS space for the univariate case. We will replace the reproducing kernel matrix R_θ by R_1 with entries $R_{1,ii'} = R_1(\mathbf{x}_i, \mathbf{x}_{i'})$, where $R_1(\cdot,\cdot)$ is defined in (1.4). The representer theorem of Kimeldorf and Wahba (1971) and the machinery of this section will follow.

As can be seen from the formulation (1.12), the SS-ANOVA model has $q+1$ smoothing parameters for a q component model. There is an overparameterization, since θ_j/λ's are equivalently representing the set; this overparameterization is mainly for computational convenience and commonly used in practice. Since for most of the time we do not know what the smoothing parameters are supposed to be, one main difficulty associated with the high dimensional SS-ANOVA models is to tune these parameters adaptively. A $(q+1)$-dimensional grid search is often required for a q-component SS-ANOVA model, which can greatly inflate the total computation time for large dimensional regression problems. We will revisit this issue in the later sections.

1.4 Component Selection for Nonparametric Models

Before introducing the component selection problem in nonparametric regression, we would like to clarify the difference between a "variable" and a "component". Any segment in the functional ANOVA decomposition of (1.7) will be called a component. In other words, each main effect, first, second or higher-order interaction effect included in the truncated functional ANOVA definition (1.9) is a component. On the other hand, if we assume an additive model, each component represents the main effect of the corresponding variable, hence a component and a variable will become synonymous.

Component (variable) selection is a more challenging task in nonparametric regression. The primary challenge is that, in nonparametric regression models, the predictive ability of a component (or a variable) cannot be represented with one parameter alone, such as a regression coefficient in linear models. For ease of discussion, we now consider an additive regression model. The extension of the following argument to models including interactions is quite straightforward.

In linear regression, the variable selection process is easier since the absolute magnitude of the regression coefficient ($\hat{\beta}_j$) measures the importance of the corresponding variable ($\mathbf{X}_j$), and defining $\hat{\beta}_j = 0$ will exclude the variable from the model. In contrast, in nonparametric regression we need to estimate the whole functional component ($\hat{f}_j$), and in order to exclude the corresponding component from the model, we have to set $\hat{f}_j$ as the *zero function*.

Because of the more difficult nature of the process in nonparametric regression, there are not as many variable selection methods available for nonparametric models in the literature as there are for their linear model counterpart. Gu (1992) proposed a set of cosine diagnostic tools to detect possible aliasing effects, in search of building a parsimonious SS-ANOVA model. The strategy is not automated, and it needs rebuilding the model and applying the diagnostics in an interactive way. A generalization of LASSO to SS-ANOVA for exponential families is proposed by Zhang et al. (2004). The method uses basis expansion of the nonparametric components with a large number of basis functions, and applies the L_1 penalty to reach a sparse solution of coefficients for these basis functions. However, the sparsity in basis functions does not guarantee the sparsity in SS-ANOVA model, since some of the coefficients in the extended linear space will not be estimated as zero, hence the variable will not entirely disappear from the model. Therefore a separate model selection has to be applied (Lin and Zhang 2006; Zhang and Lin 2006).

Multivariate adaptive regression splines (MARS - Friedman 1991) and component selection and smoothing operator (COSSO hereafter - Lin and Zhang 2006, Zhang and Lin 2006) are two popular methods for component selection in nonparametric regression settings. A short discussion on the component selection using MARS has already been discussed in Section 1.2. The COSSO (will be referred as original COSSO hereafter) is the base of our proposal, hence an introduction to this method is provided in the following section.

1.4.1 COmponent Selection and Smoothing Operator - COSSO

In this section, we shortly review the COSSO, a promising variable selection method in nonparametric regression by Lin and Zhang (2006). COSSO is a regularization method with the penalty functional being the sum of RKHS norms. The method applies a soft thresholding type operation to the function components and therefore executes variable selection and model fitting simultaneously.

The COSSO procedure proposes to find $f \in \mathscr{H}$ to minimize:

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i)]^2 + \eta \sum_{j=1}^{q} \|P^j f\|_{\mathscr{H}} \tag{1.16}$$

where η is the smoothing parameter and $\mathscr{H} = [1] \oplus_{j=1}^{q} \mathscr{H}^j$ is the same truncated functional ANOVA space as defined for SS-ANOVA model in (1.12). We will make the convention $0/0 = 0$ throughout this dissertation. The COSSO is a generalization of the LASSO's shrinkage idea into the SS-ANOVA framework. In fact, it is shown by Lin and Zhang (2006) that the LASSO can be seen as a special case of COSSO.

The COSSO method has several advantages over the traditional SS-ANOVA estimation. First, the method not only fits a nonparametric regression model to multivariate data, but it also conducts automatic component selection. The second advantage is that, the method involves only one smoothing parameter; therefore, it will not deal with the q-dimensional tuning problem as in the SS-ANOVA. This property of COSSO is very appealing for reducing the computational time even if component selection is not the main purpose.

The minimization problem of equation (1.16) is not an easy task though. An algorithm based on iterations between fitting the SS-ANOVA with fixed smoothing parameters and solving a quadratic programming (QP) is provided in Lin and Zhang (2006) and Zhang and Lin (2006). The existence of the COSSO solution was proven in the same papers. The method is compared to the MARS of Friedman (1991). In terms of both variable selection and prediction accuracy, the COSSO outperforms the MARS.

The original COSSO method works based on the assumption of independent error terms. In other words, in situations where the data is correlated (i.e., clustered, repeated measures or time series data) the performance of the COSSO method is questionable. In this dissertation research, we propose to generalize COSSO for conducting variable selection and function estimation jointly for *correlated* data. To our best knowledge, there is very little work which handles this type of problems in the literature.

1.5 Regression for Correlated Data

Almost every component selection method we introduced until now assumes independence between observations. Although this assumption might be valid for some cases, in other situations such as longitudinal, clustered or time series data sets, the responses are naturally correlated. As an illustration, we expect correlation between observations from the same subject in a longitudinal data situation, or observations belonging to the same cluster in a clustered data. In the time series setting, a common assumption is that observations closer in time have a higher correlation than observations taken further away from each other. There are cases where ignoring a possible correlation among observations will invalidate the model estimation, variable selection and statistical inferences. Although there are exceptions where the model estimation is not disturbed, even in these situations defining a proper correlation structure will improve the model efficiency. Therefore, correlation modelling is helpful to improve the efficiency of any type of estimation or component selection method.

In this research, we will focus on the situation where the error terms of the model (1.1) are correlated. In particular, we assume that the correlation matrix depends on a parsimonious set of parameters:

$$\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^{\mathrm{T}} \sim N(0, \sigma^2 \mathbf{W}_\tau^{-1}), \tag{1.17}$$

where $\mathbf{W}_\tau^{-1}$ has a known correlation structure depending on a set of unknown covariance parameters τ.

With this formulation, we are automatically making an equal variance assumption in between the error terms. In other words, we are making the assumption that $E[\varepsilon_i^2] = \sigma^2, \forall i$. This assumption can easily be relaxed by defining an error covariance such as $\mathbf{V}_\tau$, where τ includes both variance and covariance parameters. For ease of notation, we will use (1.17) throughout this research.

We now present two examples of the $\mathbf{W}_\tau^{-1}$ matrix. Consider a first-order stationary autoregressive AR(1) model for ε_i, $\varepsilon_i = \rho \varepsilon_{i-1} + a_i$, where $a_i \sim N(0, \sigma^2)$ are independent, and

$|\rho| < 1$. We then have $\tau = \rho$, and the corresponding $\mathbf{W}_\rho^{-1}$ matrix is:

$$\mathbf{W}_\rho^{-1} = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ & 1 & \rho & \dots & \rho^{n-2} \\ & & 1 & \vdots & \\ & & & & 1 \end{pmatrix}, \tag{1.18}$$

where the observations close in time will have a higher correlation compared to ones apart in time. We will come back to this covariance structure in the first simulation example.

The second example structure is useful in covariance modelling of longitudinal or clustered data, where observations belonging to the same subject (or cluster) will have correlation, and the ones not belonging to different subjects (or clusters) are assumed to be independent. This will give us a block diagonal structure for $\mathbf{W}^{-1}$:

$$\mathbf{W}^{-1} = \begin{pmatrix} \Sigma_1 & 0 & 0 & \dots & 0 \\ & \Sigma_2 & 0 & \dots & 0 \\ & & \Sigma_3 & \vdots & \\ & & & & \Sigma_m \end{pmatrix} \tag{1.19}$$

where Σ_j is an $n_j \times n_j$ square matrix, and n_j is the number of observations per subject (or cluster), and m is the total number of subjects (clusters) in the dataset.

A second step of covariance modelling is needed to specify the correlation structure within subjects. Although we might still assume an AR(1) structure within Σ_j, the assumption might not be appropriate in a clustered data setting. Alternatively, we can assume a compound symmetry (CS) within-cluster correlation structure where observations belonging to the same subject have equal correlation between each other. The corresponding Σ_j matrix then becomes:

$$\Sigma_j = (1-\rho)I_{n_j} + \rho J_{n_j} \tag{1.20}$$

where $j = 1,\ldots,m$, I_{n_j} is the identity matrix with size n_j, and J_{n_j} is the matrix of ones with size $n_j \times n_j$.

In the following section, we will cover a nonparametric regression model, to be specific, the SS-ANOVA model for correlated data situations. Reader should remark that the SS-ANOVA model for correlated data does not conduct component selection.

1.5.1 Smoothing Spline ANOVA Model for Correlated Data

In literature, the SS-ANOVA method is based on the assumption that random errors are independent. In practice, we often encounter the situations where the error terms actually are correlated. Ignoring the correlation among the error terms may result in poor performance of smoothing parameter selection and function estimation for SS-ANOVA models (see Altman 1990, Diggle and Hutchinson 1989 and Wang 1998b).

We now consider the p-variate regression problem in equation (1.1) with correlated additive error terms:

$$\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^{\mathrm{T}} \sim N(0, \sigma^2 \mathbf{W}_\tau^{-1}), \tag{1.21}$$

where σ^2 is unknown, and $\mathbf{W}_\tau^{-1}$ also depends on some unknown correlation parameters τ.

We continue the discussion from the functional ANOVA decomposition in (1.7) and the truncated RKHS $\mathscr{H}$ in (1.9). Recall that the inner product in $\mathscr{H}$ is defined in (1.10), and the Reproducing Kernel (RK) of the product space is $R_\theta(\cdot,\cdot) = \sum_{j=1}^q \theta_j R_j(\cdot,\cdot)$.

The marginal distribution of $\mathbf{y}$ is used for estimation purposes. It can easily be seen that:

$$\mathbf{y} \sim N(\mathbf{f}, \sigma^2 \mathbf{W}_\tau^{-1}).$$

For estimation, it would be natural to consider the penalized log-likelihood of $\mathbf{y}$:

$$\frac{n}{2}\log\sigma^2 - \frac{1}{2}\log|\mathbf{W}_\tau| + \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda \sum_{j=1}^q \theta_j^{-1}\|P^j f\|_{\mathscr{H}}^2. \tag{1.22}$$

Assuming $f(\mathbf{x})$ lies in $\mathscr{H}$, for fixed τ, σ^2, λ and θ, the smoothing spline ANOVA estimate of f is the minimizer of the following penalized weighted least squares problem:

$$\min_{f\in\mathscr{H}} (\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda^* \sum_{j=1}^q \theta_j^{-1}\|P^j f\|_{\mathscr{H}}^2, \tag{1.23}$$

with $\lambda^* = 2\sigma^2\lambda$. So the representer theorem of Kimeldorf and Wahba (1971) can be applied to the correlated SS-ANOVA situation as well, when we fix the smoothing and variance estimators. Therefore, the estimation of the SS-ANOVA model is equivalent to solving $\hat{\mathbf{c}}$ and $\hat{d}$, where $\hat{\mathbf{f}} = T\hat{d} + R_\theta \hat{\mathbf{c}}$, where the $n \times n$ matrix R_θ with entries $R_{\theta,ii'} = R_\theta(\mathbf{x}_i, \mathbf{x}_{i'})$, and $R_\theta(\cdot,\cdot)$ is defined in (1.11). Although the calculation of $\hat{\mathbf{c}}$ and $\hat{d}$ goes parallel to the discussion in Section 1.3.3, slight differences will be observed since $\mathbf{W}_\tau^{-1}$ is involved in the estimation:

$$\begin{aligned} \left(R_\theta + n\lambda^* \mathbf{W}_\tau^{-1}\right)\mathbf{c} + \mathbf{1}d &= \mathbf{y}, \\ \mathbf{1}^{\mathrm{T}}\mathbf{c} &= \mathbf{0}. \end{aligned} \tag{1.24}$$

After some algebra, the following matrix operations provide the estimates:

$$\begin{aligned} \hat{\mathbf{c}} &= Q_2 \left[Q_2^{\mathrm{T}}(R_\theta + n\lambda^* \mathbf{W}_\tau^{-1})Q_2\right]^{-1} Q_2^{\mathrm{T}}\mathbf{y}, \\ \hat{d} &= S^{-1}\left[Q_1^{\mathrm{T}}(\mathbf{y} - (R_\theta + n\lambda^* \mathbf{W}_\tau^{-1})\mathbf{c})\right], \end{aligned}$$

where Q_1, Q_2 and S are from the QR decomposition of $\mathbf{1}$ in Section 1.3.3. The corresponding *hat* matrix ($\mathbf{A}$) can be calculated as:

$$\mathbf{A} = I - n\lambda^* \mathbf{W}_\tau^{-1} Q_2 \left[Q_2^{\mathrm{T}}(R_\theta + n\lambda^* \mathbf{W}_\tau^{-1})Q_2\right]^{-1} Q_2^{\mathrm{T}}.$$

Note that $\mathbf{A}$ is not necessarily symmetric anymore as it was for independent errors scenario.

Careful readers should remark that the computation process given above depends on the knowledge of τ, σ, θ and λ^*. As opposed to the SS-ANOVA model with independent error terms, these parameters should be either known, or estimated from data simultaneously (Wang, 1998b; Gu and Han, 2004; Opsomer, Wang, and Yang, 2001). Most of the time we do not have information either on the variance-covariance parameters (τ, σ) or on the smoothing parameters (λ^*, θ), and therefore a method to estimate these parameters along with $f(\mathbf{x})$ is needed. This brings an extra challenge in solving (1.23), and the method in the following section gives a feasible solution to this issue.

1.5.2 Generalized Maximum Likelihood

It is well known from the smoothing spline literature that ignoring the correlation in error terms will affect the performance of both function estimation and parameter tuning (Wang, 1998b; Opsomer, Wang, and Yang, 2001; Gu and Han, 2004). Several methods have been developed to select smoothing parameters and covariance parameters jointly, among which Wang (1998b) and Opsomer, Wang, and Yang (2001) extends the Generalized Maximum Likelihood (GML) approach of Wahba (1985) to perform the joint estimation.

Consider the following Bayesian model as a prior distribution for f:

$$F(\mathbf{x}) = \gamma + b^{1/2} \sum_{i=1}^{q} \theta_j^{1/2} U_j(\mathbf{x}), \quad \mathbf{x} \in \mathscr{T}, \tag{1.25}$$

where $\gamma \sim N(0, a)$, a and $b = \frac{\sigma^2}{n\lambda^*}$ are positive constants. $U_j(\mathbf{x}), \mathbf{x} \in \mathscr{T}$ is a zero mean Gaussian stochastic process independent of γ with the covariance $U_j(\mathbf{x}_i)U_j(\mathbf{x}_k) = R_j(\mathbf{x}_i, \mathbf{x}_k)$. Define $\mathbf{y}$ with the additive correlated error terms:

$$y_i = F(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}} \sim N(\mathbf{0}, \sigma^2 \mathbf{W}_\tau^{-1})$. The marginal distribution of $\mathbf{y}$ is $N(0, b(\eta \mathbf{1}\mathbf{1}^{\mathrm{T}} + R_\theta + n\lambda^* \mathbf{W}_\tau^{-1}))$ with $\eta = a/b$.

It can be shown that, when a approaches to infinity, the posterior mean of the process is the smoothing spline estimate (Wahba 1990, Opsomer, Wang, and Yang 2001, Wang 1998b), i.e.,

$$\lim_{a \to \infty} \mathrm{E}\left(F(\mathbf{x})|\mathbf{y}\right) \quad = \quad \hat{f}(\mathbf{x}).$$

Define the following contrasts of the $\mathbf{y}$ vector:

$$\begin{pmatrix} \mathbf{z} \\ \\ w \end{pmatrix} = \begin{pmatrix} Q_2^{\mathrm{T}} \\ \\ \frac{1}{\sqrt{\eta}} \mathbf{1}^{\mathrm{T}} \end{pmatrix} \mathbf{y},$$

where Q_2 is defined in the QR decomposition of $\mathbf{1}$ previously in this chapter. Remark that w is asymptotically uncorrelated with $\mathbf{z}$, and the distribution of w is independent of $\lambda^*, \theta, \sigma^2$ and τ (see Wang 1998b). The maximum likelihood estimates of these parameters can be based on the marginal distribution of $\mathbf{z}$,

$$\mathbf{z} \sim N\left(\mathbf{0}, b Q_2^{\mathrm{T}} \left(\eta R_\theta + n\lambda^* \mathbf{W}_\tau^{-1}\right) Q_2\right).$$

Let $\mathbf{B}(\lambda, \theta, \tau) = \left(\eta R_\theta + n\lambda^* \mathbf{W}_\tau^{-1}\right)$. Then the Generalized Maximum Likelihood (GML) estimators of $\lambda^*, \theta, \sigma^2$ and τ are based on maximizing the marginal log-likelihood of $\mathbf{z}$:

$$l(\tau, \sigma^2, \lambda^*, \theta|\mathbf{z}) = -\frac{1}{2}\log\left|\frac{\sigma^2}{n\lambda^*}Q_2^{\mathrm{T}}\mathbf{B}(\lambda^*, \theta, \tau)Q_2\right| - \frac{n\lambda^*}{2\sigma^2}\mathbf{z}^{\mathrm{T}}\left(Q_2^{\mathrm{T}}\mathbf{B}(\lambda^*, \theta, \tau)Q_2\right)^{-1}\mathbf{z} + C$$

where C is constant. Maximizing the likelihood above with respect to σ^2 gives

$$\hat{\sigma}^2 = \frac{n\lambda^*\mathbf{z}^{\mathrm{T}}\left(Q_2^{\mathrm{T}}\mathbf{B}(\lambda^*, \theta, \tau)Q_2\right)^{-1}\mathbf{z}}{n-1}, \tag{1.26}$$

and the GML estimates of λ^*, θ and τ are the minimizers of

$$M(\tau, \theta, \lambda^*) = \frac{\mathbf{z}^{\mathrm{T}}\left(Q_2^{\mathrm{T}}\mathbf{B}(\lambda^*, \theta, \tau)Q_2\right)^{-1}\mathbf{z}}{[det\left(Q_2^{\mathrm{T}}\mathbf{B}(\lambda^*, \theta, \tau)Q_2\right)^{-1}]^{\frac{1}{n-1}}} = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{W}_\tau(I-\mathbf{A})\mathbf{y}}{[det^+\left(\mathbf{W}_\tau(I-\mathbf{A})\right)]^{\frac{1}{n-1}}}, \tag{1.27}$$

where det^+ is the product of the nonzero eigenvalues.

1.5.3 Linear Mixed Models Representation

Recall that in 1.5.2, we estimate the functional components $(\hat{f})$ by minimizing the penalized weighted least squares, and use the GML method to estimate covariance parameters (τ, σ) and the smoothing parameters (λ^*, θ) simultaneously. The main idea of this section is to use the connection between SS-ANOVA estimates and the linear mixed models, and get benefit from the well studied theory and computational power of linear mixed models for this estimation. We would like to introduce the connection between the smoothing spline ANOVA model and the linear mixed effects model. The connection was first pointed out by Speed in the discussion of Robinson (1991), and later studied thoroughly in Wang (1998b).

Define $Z = (I_n, \cdots, I_n)$, the q copies of the identity matrix of size n, here n is the number of observations and with some abuse of notation $R_j = \{R_j(\mathbf{x}_i, \mathbf{x}_{i'})\}$, $i, i' = 1, \ldots, n$. Consider the following linear mixed model:

$$\mathbf{y} = \mathbf{1}d + \sum_{j=1}^{q} \mathbf{u}_j + \varepsilon = \mathbf{1}d + Z\mathbf{u} + \varepsilon, \tag{1.28}$$

where d is the fixed intercept effect, $\mathbf{u}_j$'s are random effects $\mathbf{u}_j \sim N\left(\mathbf{0}, \sigma^2\theta_j R_j/(n\lambda^*)\right)$, ε's are error term with $\varepsilon \sim N(\mathbf{0}, \sigma^2\mathbf{W}_\tau^{-1})$, and $\mathbf{u}_j$'s and ε are mutually independent. Define $\mathbf{u} = (\mathbf{u}_1^{\mathrm{T}}, \cdots, \mathbf{u}_q^{\mathrm{T}})^{\mathrm{T}}$, and $D = \mathrm{diag}(\theta_1 R_1, \cdots, \theta_q R_q)$. Then $Cov(\mathbf{u}) = \sigma^2 D/n\lambda^*$. Also define $\mathbf{u} = D\phi$. In the mixed effects model, the matrix representation to solve for the fixed effect $\hat{d}$ and the random effects $\hat{\mathbf{u}}$ can be obtained by using Harville's generalized equations (Harville, 1977).

$$\begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau\mathbf{1} & \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau ZD \\ DZ^{\mathrm{T}}\mathbf{W}_\tau\mathbf{1} & DZ^{\mathrm{T}}\mathbf{W}_\tau ZD + n\lambda^* D \end{pmatrix} \begin{pmatrix} \hat{d} \\ \hat{\phi} \end{pmatrix} = \begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau\mathbf{y} \\ DZ^{\mathrm{T}}\mathbf{W}_\tau\mathbf{y} \end{pmatrix}. \tag{1.29}$$

It is easy to check the equation (1.29) is the same as the equation system (1.24). The estimate of $\mathbf{u}$ is $\hat{\mathbf{u}} = D\hat{\phi} = DZ^{\mathrm{T}}\hat{\mathbf{c}}$. Rephrasing the equation gives $\hat{\mathbf{u}}_j = \theta_j R_j \hat{\mathbf{c}}$ for every $j = 1, \ldots, q$. In other words, the solution of SS-ANOVA model $\hat{f} = \mathbf{1}\hat{d} + R_\theta\hat{\mathbf{c}}$ can be obtained by fitting the mixed effects model (1.28), and the estimate is a Best Linear Unbiased Predictor (BLUP, Robinson 1991). Furthermore, the GML estimates of covariance and smoothing parameters (here σ^2, τ and λ^*) are also Restricted Maximum Likelihood (REML) estimates, since these estimates are based on $n-1$ independent contrasts of $\mathbf{y}$. Reader should remark that the resulting variance component estimators do not depend on particular choice of the $(n-1)$ contrasts (Harville, 1977; Diggle, Liang, and Zeger, 2002; Verbeke and Molenberghs, 2000). The advantage with the mixed model approach is that the smoothing parameters (λ and θ's) are treated as a variance component and estimated simultaneously with σ^2 and τ; therefore, there will not be any need for q-dimensional tuning, which automatically eliminates the grid search for this parameter. Also the existing software like SAS can be used to estimate the function and parameters altogether.

The performance of the linear mixed model approach of SS-ANOVA fit is promising in low dimensions. However a big caveat about this approach is that, it requires a large number of random effects to be predicted. Looking at the model (1.28) more carefully, the model fitting requires the prediction of nq random effects, where n is the number of observations, and q is the number of components in SS-ANOVA. The number of random effects increases rapidly as the dimension of the data increases. As an illustration, if we have a data set with 400 observations and 30 explanatory variables, the simplest additive model fit using the mixed model connection will require a prediction of $12,000$ random effects. It easily takes a lot of computer memory and computation time for optimization in the case of a large number of random effects. Therefore, the estimation becomes infeasible, or at least very slow for high dimensional data.

The mixed model connection helps to solve the SS-ANOVA model, but it does not conduct component selection. As mentioned earlier, most of the nonparametric component selection methods (including MARS and original COSSO) assume independent error terms. In this dissertation research, we propose the new method Correlated Component Selection and Smoothing Operator (Cor-COSSO), which intends to apply simultaneous component selection and model fitting in the SS-ANOVA framework for correlated data. Our new method is formulated in a way that the computation time will not be affected by the number of explanatory variables. This is an advantage compared to other nonparametric regression methods which require multidimensional tuning. Therefore, our method can also be considered as an alternative approach to tackle the multidimensional smoothing parameter selection problem. In other words, even if the main purpose of the analysis is not component selection but fitting high-dimensional nonparametric regression for correlated data, our new method still offers an alternative estimation approach which does not suffer much from high dimensionality.

1.6 Dissertation Outline

We propose a nonparametric component selection method working with correlated error terms. The method, Correlated COSSO, is an extension of COSSO to correlated data situations such as time series, clustered or longitudinal data. Correlated COSSO conducts component selection, model fitting, variance-covariance parameter estimation and smoothing parameter tuning simultaneously. For computation, our method takes advantage of the connection between the smoothing spline ANOVA and linear mixed effects models, and benefits from available commercial software for mixed models such as *SAS Proc Mixed*. An efficient algorithm is developed for optimization, which solves the smoothing spline with correlated data and a quadratic programming (QP) iteratively. The rest of the dissertation is organized as follows.

Chapter 2 introduces the Correlated COSSO method and develops an efficient algorithm for optimization. The existence of the solution is proven. The solution is also shown to have a finite dimensional representation. Smoothing parameter selection issue is raised, and several alternative methods are discussed.

Chapter 3 proposes an extension of the method, the Adaptive Correlated COSSO, and studies its properties. The method is announced as a further improvement on Correlated COSSO method. The computation algorithms and smoothing parameter selection issues are discussed.

The computation issue for large datasets is handled in Chapter 4. We consider a method called the subset basis algorithm to reduce the computational time of the Correlated COSSO and the Adaptive Correlated COSSO methods in massive datasets. The algorithm uses a subset of the dataset to find an approximate set of basis functions. A discussion on different sampling methods to select the subset is included in this chapter as well.

In Chapter 5, the empirical performance of both Correlated COSSO and Adaptive Correlated COSSO methods are evaluated with extensive simulation studies. The chapter consists of two parts; the first part contains examples to illustrate different tuning criteria, computational algorithms and the subset technique, and the second part is designed to compare the two proposals with other existing methods from the literature.

In Chapter 6 we implemented the Correlated COSSO and Adaptive Correlated COSSO to two real data examples. The example data sets are selected to show the wide range of application possibilities of both methods. The first example is Ozone data (Breiman and Friedman, 1985) where 8 meteorological variables are used to model the daily maximum ozone readings in Los Angeles basin. The second example is a Money Demand study where 4 explanatory variables are used to model the log-log demand, which is measured by the real money stock.

Chapter 7 gives a short conclusion and discusses the future work.

CHAPTER 2

CORRELATED COMPONENT SELECTION AND SMOOTHING OPERATOR (COR-COSSO)

2.1 Introduction

In this dissertation research, our main purpose is to propose a nonparametric component selection method suitable for correlated data. In nonparametric regression, it is well known that the correlation in data affects the performances of tuning methods (Wang 1998b, Gu and Han 2004). To handle this issue, the joint estimation of variance-covariance parameters and smoothing parameters is recommended. However, the problem becomes more complicated when variable selection is also involved. The popular nonparametric component selection methods such as MARS and COSSO work mainly under the independent error assumption. To our knowledge, very little work on nonparametric component selection methods which take the correlation into account exists in the literature.

We propose a generalization of COSSO (Lin and Zhang, 2006; Zhang and Lin, 2006) by taking the correlation into account. The new method is called the Correlated COSSO (Cor-COSSO hereafter), which simultaneously conducts model estimation, component selection, smoothing and covariance parameters estimation. In particular, the Cor-COSSO solves the penalized weighted least squares problem, with a soft-thresholding penalty on the functional components which encourages sparse estimation, while taking data covariance structure into account. The COSSO penalty functional is the sum of component norms instead of the squared norms as it is in SS-ANOVA method.

In Section 2.2 of this chapter, we describe the formulation of our method in details. The existence of the solution to Cor-COSSO minimization problem is proven. In order to facilitate the computation of the method, we present an alternative formulation which leads to an efficient iterative algorithm to solve Correlated COSSO problem.

We propose four algorithms to solve the optimization problem associated with Cor-COSSO in Section 2.3; three are based on the full iteration (including some variations), and one is based on the one-step iteration. All algorithms use two *stages*: ESTIMATION and SELECTION. The full iteration algorithms iterate between these two stages until convergence, while the one-step algorithm solves both stages only once. A simulation study will be provided in Section 5.2.2 to compare the effectiveness of these algorithms.

As in many nonparametric regression problems, the selection of the smoothing parameter is crucial in Cor-COSSO. Section 2.3.2 is devoted to the discussion of smoothing parameter selection. We cover a variety of tuning methods for the selection of tuning parameters. These methods will be compared in Section 5.2.1 with a simulation study.

2.2 Cor-COSSO: Method and Formulation

Consider the p-variate regression problem:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, ..., n, \tag{2.1}$$

with the additive error terms

$$\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^{\mathrm{T}} \sim N(0, \sigma^2 \mathbf{W}_\tau^{-1}), \tag{2.2}$$

where $\mathbf{W}_\tau^{-1}$ has a known correlation structure depending on a set of correlation parameters τ. Define $\mathbf{f} = \Big(f(x_1), f(x_2), \ldots, f(x_n)\Big)^{\mathrm{T}}$ and $\mathbf{y} = \Big(y_1, y_2, \ldots, y_n\Big)^{\mathrm{T}}$. It is natural to consider the penalized log likelihood of $\mathbf{y}$:

$$\frac{n}{2} \log \sigma^2 - \frac{1}{2} \log |\mathbf{W}_\tau| + \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{f})^{\mathrm{T}} \mathbf{W}_\tau (\mathbf{y} - \mathbf{f}) + n\lambda J(f), \tag{2.3}$$

where λ is a smoothing parameter, and $\mathbf{W}_\tau$ is the inverse of error correlation matrix, and the $J(f)$ is the penalty term. In the Correlated COSSO method, we use the penalty term $J(f) = \sum_{j=1}^{q} \|P^j f\|$ which is a sum of RKHS norms instead of the squared norms used in SS-ANOVA.

With fixed σ^2, τ, λ, Correlated COSSO method proposes to find $f \in \mathscr{H}$ by minimizing the penalized weighted least squares:

$$\min_{f\in\mathscr{H}} \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_{\tau}(\mathbf{y}-\mathbf{f})+n\lambda\sum_{j=1}^{q}\|P^j f\|. \tag{2.4}$$

With regard to the variance covariance parameters (σ^2,τ) and the smoothing parameter (λ), we propose to estimate them using Generalized Maximum Likelihood (GML) method. One important feature of the Correlated COSSO is that, no matter how many components are included in the model, the Correlated COSSO method has only one smoothing parameter. As mentioned earlier, high dimensional tuning of smoothing parameters is a bottleneck for the standard SS-ANOVA models. Here, Cor-COSSO overcomes this difficulty by its formulation.

2.2.1 Existence of Cor-COSSO Estimate

In the following part, we focus on (2.4), which assumes σ^2,τ and λ are fixed. The existence of the Correlated COSSO estimate is guaranteed by the following theorem.

Theorem 2.2.1 *Let $\mathscr{H}$ be an RKHS of functions over an input space $\mathscr{T}$. Assume that $\mathscr{H}$ can be decomposed as:*

$$\mathscr{H}=[1]\oplus\mathscr{H}_1 \quad \textit{with} \quad \mathscr{H}_1=\oplus_{j=1}^{q}\mathscr{H}^j.$$

Then there exists a minimizer of (2.4) *in $\mathscr{H}$.*

The proof of Theorem 2.2.1 will be included here for the completeness of the section. However, before going through the proof, we will need some extra notations and definitions.

Notations and Definitions

Let $R_{\mathscr{H}_1}$ be the reproducing kernel of $\mathscr{H}_1$ and $< \cdot,\cdot >_{\mathscr{H}_1}$ be the inner product in $\mathscr{H}_1$. Let $a = \Big\{ \sum_{i=1}^{n} R_{\mathscr{H}_1}(x_i,x_i) \Big\}^{1/2}$. Define the eigen decomposition of $\mathbf{W}$ as $\mathbf{W} = \mathbf{LOL}^{\mathrm{T}}$ where $\mathbf{O} = diag\big(d_1,\ldots,d_n\big), \mathbf{L} : \mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{LL}^{\mathrm{T}} = I$.

Let $\tilde{d} = \min_{i=1}^{n}\{d_i\}$, then $\tilde{d} > 0$. Let $z_i = \sum_{j=1}^{n} l_{ji}y_j$ for $i = 1,2,\ldots,n$, where l_{ji} is the ji^{th} element of matrix $\mathbf{L}$, and $\tilde{l} = \min_{i=1}^{n} \Big| \sum_{j=1}^{n} l_{ji} \Big|$. Define $\varsigma = \max_{i=1}^{n} \Big\{ d_i z_i^2 + |z_i| + 1 \Big\}$.

We need the following two lemmas and a theorem from Tapia and Thompson (1978) in order to prove Theorem 2.2.1. We will state the lemmas here and include the proofs at the end of this section. The theorem from Tapia and Thompson (1978) will be included for the completeness, however, the proof will be omitted.

LEMMA 2.2.1 *Define* $\mathbf{f} = \Big(f(x_1), f(x_2), \ldots, f(x_n)\Big)^T$ *and* $\mathbf{y} = \Big(y_1, y_2, \ldots, y_n\Big)^T$. *Let* $A(f) = L(\mathbf{f}) + J(f)$ *where* $L(\mathbf{f}) = \frac{1}{n}\Big(\mathbf{y} - \mathbf{f}\Big)^T \mathbf{W} \Big(\mathbf{y} - \mathbf{f}\Big)$ *and* $J(f) = \sum_{j=1}^{q} \|P^j f\|$.

Then $A(f)$ *is convex and continuous.*

LEMMA 2.2.2 *Define* $\mathscr{D}$ *as following;*

If $\tilde{l} = 0$ *then* $\mathscr{D} = \Big\{ f \in \mathscr{H} : f = b + f_1, b \in [1], f_1 \in \mathscr{H}_1, J(f) \le \varsigma \Big\}$,

If $\tilde{l} > 0$ *then* $\mathscr{D} = \Big\{ f \in \mathscr{H} : f = b + f_1, b \in [1], f_1 \in \mathscr{H}_1, J(f) \le \varsigma, |b| \le \frac{1}{\tilde{l}}\{\frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1 + a)\varsigma\} \Big\}$.

Then $\mathscr{D}$ *is closed, bounded and convex.*

Theorem 2.2.2 *Theorem 4 from Tapia and Thompson (1978)*

Define the problem as:

$$\textit{minimize} \quad A(f), \quad \textit{subject to} \quad f \in \mathscr{S}$$

such that the functional $A(f)$ is convex and continuous, and $\mathscr{S}$ is closed, bounded and convex. Then the problem has at least one minimizer in $\mathscr{S}$.

Now we can proceed with the proof of Theorem 2.2.1.

Proof of Theorem 2.2.1

Let $A(f) = \frac{1}{2n\sigma^2}\Big(\mathbf{y}-\mathbf{f}\Big)^{\mathrm{T}}\mathbf{W}\Big(\mathbf{y}-\mathbf{f}\Big) + \lambda \sum_{j=1}^{q} \|P^j f\|$.

WLOG, let $\lambda = 1$ and $\sigma^2 = \frac{1}{2}$. Remark that:

$\sum_{j=1}^{q} \|P^j f\|^2 \leq \Big\{\sum_{j=1}^{q} \|P^j f\|\Big\}^2 = J(f)^2 \Rightarrow J(f) \geq \|f\| \quad \forall f \in \mathscr{H}$.

Let l_{ji} be the ji^{th} element of matrix $\mathbf{L}$ defined previously in the eigen-decomposition of $\mathbf{W}$. Using the definition of reproducing kernel:

$$\begin{aligned}
\Big|\sum_{j=1}^{n} f_1(x_j) l_{ji}\Big| &\leq |\sum_{j=1}^{n} \langle f_1(\cdot), R_{\mathscr{H}_1}(x_j,\cdot)\rangle_{\mathscr{H}_1} l_{ji}| \\
&\leq \sqrt{\sum_{j=1}^{n} \langle f_1(\cdot), R_{\mathscr{H}_1}(x_j,\cdot)\rangle^2_{\mathscr{H}_1}} \sqrt{\sum_{j=1}^{n} l_{ji}^2} \\
&\leq \|f_1\| \sqrt{\sum_{j=1}^{n} R_{\mathscr{H}_1}(x_j,x_j)} \\
&\leq aJ(f),
\end{aligned}$$

where a is defined as earlier $a = \left\{\sum_{j=1}^{n} R_{\mathscr{H}_1}(x_j,x_j)\right\}^{1/2}$, and $\langle\cdot,\cdot\rangle_{\mathscr{H}_1}$ represents the inner product of $\mathscr{H}_1$. In second line of the equation above, we used the fact that $\sum_{j=1}^{n} l_{ji}^2 = 1$ since $\mathbf{LL}^{\mathrm{T}} = I$.

Now, define the set $\mathscr{D}$ as: if $\tilde{l} = 0$ then $\mathscr{D} = \Big\{f \in \mathscr{H} : f = b + f_1, b \in [1], f_1 \in \mathscr{H}_1, J(f) \leq \varsigma\Big\}$, else if $\tilde{l} > 0$ then $\mathscr{D} = \Big\{f \in \mathscr{H} : f = b + f_1, b \in [1], f_1 \in \mathscr{H}_1, J(f) \leq \varsigma, |b| \leq \frac{1}{\tilde{l}}\{\frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1+a)\varsigma\}\Big\}$. From Lemma 2.2.2, in either case $\mathscr{D}$ is closed, convex and bounded, and from Lemma 2.2.1, $A(f)$ is convex and continuous. Following Theorem 4 of Tapia and Thompson (1978), there exists a minimizer of $A(f)$ in $\mathscr{D}$. Lets call this minimizer as $\bar{f}$.

Until this point, we have shown that there exists *a* minimizer of $A(f)$ in $\mathscr{D}$, however, we need to show this is *the* minimizer in $\mathscr{H}$ as well. In other words, we still need to show that $A(\bar{f}) < A(f)$ for any $f \in \mathscr{H} - \mathscr{D}$.

Remark that $\{0\} \in \mathscr{D}$. Then $A(\bar{f}) \leq A(0) = \frac{1}{n}\sum_{i=1}^{n} d_i z_i^2 < \varsigma$.

For any $f \in \mathscr{H}$ and $J(f) > \varsigma$, obviously $A(f) \geq J(f) > \varsigma > A(\bar{f})$. If $J(f) \leq \varsigma$ and $|b| > \frac{1}{\tilde{l}}K$, where $K = \frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1+a)\varsigma$, then for each $i = 1,2,\ldots,n$:

$$\begin{aligned}
\sqrt{d_i}\Big|z_i - b\Big(\sum_{j=1}^{n} l_{ji}\Big) - \sum_{j=1}^{n} l_{ji} f_1(x_j)\Big| &\geq \sqrt{d_i}\left(\Big|b\Big|\Big|\sum_{j=1}^{n} l_{ji}\Big| - \Big|z_i\Big| - \Big|\sum_{j=1}^{n} l_{ji} f_1(x_j)\Big|\right) \\
&> \sqrt{d_i}\left(\Big\{\frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1+a)\varsigma\Big\}\frac{\Big|\sum_{j=1}^{n} l_{ji}\Big|}{\tilde{l}} - (1+a)\varsigma\right) \\
&\geq \sqrt{d_i}\left(\frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1+a)\varsigma - (1+a)\varsigma\right) \\
&= \sqrt{\frac{d_i}{\tilde{d}}}\sqrt{\varsigma} \geq \sqrt{\varsigma}.
\end{aligned}$$

In the inequalities above, the first line uses $\left|a-b-c\right| \geq \left|a\right| - \left|b\right| - \left|c\right|$, while going from the first to second line, we use $\left|b\right| > \frac{K}{\tilde{l}}$, $\left|z_i\right| < \varsigma$ and $\left|\sum_{j=1}^{n} l_{ji} f(x_j)\right| \leq aJ(f) \leq a\varsigma$. The third line follows from $\frac{\left|\sum_{j=1}^{n} l_{ji}\right|}{\tilde{l}} \geq 1$, and the fourth line follows from $\frac{d_i}{\tilde{d}} \geq 1$.

Consider $A(f)$ now:

$$
\begin{aligned}
A(f) \geq L(\mathbf{f}) \quad &= \quad \frac{1}{n}\Big(\mathbf{y}-\mathbf{f}\Big)^{\mathsf{T}}\mathbf{W}\Big(\mathbf{y}-\mathbf{f}\Big) \\
&= \quad \frac{1}{n}\Big(\mathbf{y}-\mathbf{f}\Big)^{\mathsf{T}}\mathbf{LOL}^{\mathsf{T}}\Big(\mathbf{y}-\mathbf{f}\Big) \\
&= \quad \frac{1}{n}\sum_{i=1}^{n}\left\{d_i\left(\sum_{j=1}^{n} l_{ji}y_j - b\Big(\sum_{j=1}^{n} l_{ij}\Big) - \sum_{j=1}^{n} l_{ij}f_1(x_j)\right)^2\right\} \\
&\geq \quad \min_{i=1}^{n}\left\{d_i\left(z_i - b\Big(\sum_{j=1}^{n} l_{ij}\Big) - \sum_{j=1}^{n} l_{ij}f_1(x_j)\right)^2\right\} \\
&\geq \quad \sqrt{\varsigma}^2 = \varsigma \\
&> \quad A(\bar{f}).
\end{aligned}
$$

Here we use the $\mathbf{W} = \mathbf{LOL}^{\mathsf{T}}$ decomposition passing from the first line to second and decompose the matrix format for the third line. The forth line uses the fact that the average of positive numbers will be greater than or equal to the minimum of the same set of numbers. Since every $d_i > 0$, we can write this inequality. The fifth line uses the inequality we just showed above. Therefore, $\bar{f}$ is the minimizer of $A(f)$ in $\mathscr{H}$, hence the proof holds.

2.2.2 Finite Dimensional Representation of Cor-COSSO Estimate

Theorem 2.2.1 guarantees that there exists a solution to (2.4). Now we would like to show that the solution can be estimated in a finite dimensional space, i.e., there exists a finite dimensional representation of the solution. The following theorem proves that, for any fixed σ^2, τ, λ, the solution to (2.4) lies in a finite dimensional space. This result is parallel to the representer theorem from smoothing splines (Kimeldorf and Wahba, 1971).

Theorem 2.2.3 *For any fixed τ, σ^2 and λ, let the minimizer of* (2.4) *be $\hat{f} = \hat{d} + \sum_{j=1}^{q} \hat{f}_j$, with $\hat{f}_j \in \mathscr{H}^j$. Then $\hat{f}_j \in span\{R_j(\mathbf{x}_i, \cdot), i = 1, ..., n\}$, where $R_j(\cdot, \cdot)$ is the reproducing kernel of $\mathscr{H}^j$.*

Proof of Theorem 2.2.3 Define the reparameterized objective function of (2.4) as $A(f) = \frac{1}{n}(\mathbf{y} - \mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y} - \mathbf{f}) + \lambda^* \sum_{j=1}^{q} \|P^j f\|$ with $\lambda^* = 2\sigma^2\lambda$. For any $f \in \mathscr{H}$, we can write $f = d + \sum_{j=1}^{q} f_j$ with $f_j \in \mathscr{H}^j$. Let the projection of f_j onto $\text{span}\Big\{R_j\big(\mathbf{x}_i, \cdot\big), i = 1, ..., n\Big\} \subset \mathscr{H}^j$ be denoted by g_j, and the orthogonal complement by h_j. Then:

$$f_j = g_j + h_j, \quad \text{and} \quad \|f_j\|^2 = \|g_j\|^2 + \|h_j\|^2, \quad j = 1 \ldots, q.$$

Define the inner product in $\mathscr{H}$ as $\langle \cdot, \cdot \rangle_{\mathscr{H}}$. Remark that the reproducing kernel for $\mathscr{H}$ is $R(\cdot,\cdot) = 1 + \sum_{j=1}^{q} R_j(\cdot,\cdot)$, and:

$$
\begin{aligned}
f(\mathbf{x}_i) = \langle R(\mathbf{x}_i, \cdot), f(\cdot) \rangle &= \langle 1 + \sum_{j=1}^{q} R_j, d + \sum_{j=1}^{q} f_j \rangle_{\mathscr{H}} \\
&= \langle 1 + \sum_{j=1}^{q} R_j, d + \sum_{j=1}^{q} \{g_j + h_j\} \rangle_{\mathscr{H}} \\
&= \langle 1, d \rangle_{\mathscr{H}} + \sum_{j=1}^{q} \langle 1, \{g_j + h_j\} \rangle_{\mathscr{H}} + \sum_{j=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), d \rangle_{\mathscr{H}} \\
&\quad + \sum_{j=1}^{q} \sum_{k=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), g_k \rangle_{\mathscr{H}} + \sum_{j=1}^{q} \sum_{k=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), h_k \rangle_{\mathscr{H}} \\
&= d + 0 + 0 + \sum_{j=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), g_j \rangle_{\mathscr{H}} + 0 \\
&= d + \sum_{j=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), g_j \rangle_{\mathscr{H}}
\end{aligned}
$$

Therefore, we can re-express the objective function $A(f)$ as:

$$
\frac{1}{n}\Big(\mathbf{y} - d\mathbf{1} - \sum_{j=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), g_j \rangle_{\mathscr{H}}\Big)^{\mathrm{T}} \mathbf{W}_\tau \Big(\mathbf{y} - d\mathbf{1} - \sum_{j=1}^{q} \langle R_j(\mathbf{x}_i, \cdot), g_j \rangle_{\mathscr{H}}\Big) + \lambda^* \sum_{j=1}^{q} \Big\{ \|g_j\|^2 + \|h_j\|^2 \Big\}^{1/2}.
$$

The right part of the equation is strictly positive, and the h_j is not included in the left part. Hence, any minimizing f satisfies $h_j = 0 \forall j = 1, \ldots, q$, and the proof follows.

2.2.3 Equivalent Formulation

We have shown with Theorem 2.2.1 that the solution to Cor-COSSO exists, and with Theorem 2.2.3 that the solution has a finite dimensional representation. It is possible to directly compute the solution to (2.4) using this theorem. However, the optimization in this formulation is a hard minimization problem. Next, we present an equivalent formulation of (2.4), which will lead to an iterative algorithm.

Define $\theta = (\theta_1, ..., \theta_q)^{\mathrm{T}}$ and let $\mathbf{0}$ be the vector of zeros. Consider the penalized weighted least squares objective function with the new penalty term:

$$\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda_0 \sum_{j=1}^{q} \theta_j^{-1}\|P^j f\|^2 + n\lambda_1 \sum_{j=1}^{q} \theta_j, \tag{2.5}$$

where $\lambda_0 > 0$ is a constant and λ_1 is a smoothing parameter. The following lemma shows that (2.4) and (2.5) are two equivalent formulations for the Correlated COSSO.

LEMMA 2.2.3 *Set $\lambda = 2\sqrt{\lambda_0\lambda_1}$. If $\hat{f}$ minimizes* (2.4)*, set $\hat{\theta}_j = \sqrt{\frac{\lambda_0}{\lambda_1}}\|P^j\hat{f}\|$, and then $(\hat{\theta}, \hat{f})$ minimizes* (2.5)*. On the other hand, if $(\hat{\theta}, \hat{f})$ minimizes* (2.5)*, then $\hat{f}$ minimizes* (2.4)*. Therefore, solving* (2.4) *and solving* (2.5) *are equivalent.*

Proof of Lemma 2.2.3

Denote the functional in (2.4) as $D(f) = \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda\sum_{j=1}^{q}\|P^j f\|$, and the functional in (2.5) as $B(f,\theta) = \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda_0\sum_{j=1}^{q}\theta_j^{-1}\|P^j f\|^2 + n\lambda_1\sum_{j=1}^{q}\theta_j$. We need to show that:

$$\lambda\sum_{j=1}^{q}\|P^j f\| = \lambda_0\sum_{j=1}^{q}\theta_j^{-1}\|P^j f\|^2 + \lambda_1\sum_{j=1}^{q}\theta_j$$

to complete the proof.

We will use the inequality $a+b \geq 2\sqrt{ab}$ for any $a,b \geq 0$, and equality holds if and only if $a=b$ in this proof. Hence, in our case, set $a = \lambda_0 \sum_{j=1}^{q} \theta_j^{-1} \|P^j f\|^2$, and $b = \lambda_1 \sum_{j=1}^{q} \theta_j$. Then, $a+b = \lambda \sum_{j=1}^{q} \|P^j f\| = 2\sqrt{\lambda_0 \sum_{j=1}^{q} \theta_j^{-1} \|P^j f\|^2}\sqrt{\lambda_1 \sum_{j=1}^{q} \theta_j}$. For each $j = 1,\ldots,q$:

$$\begin{aligned} \lambda \|P^j f\| &= 2\sqrt{\lambda_0 \theta_j^{-1} \|P^j f\|^2}\sqrt{\lambda_1 \theta_j} \\ &= 2\sqrt{\lambda_0 \lambda_1 \theta_j^{-1} \theta_j \|P^j f\|^2} \\ &= 2\sqrt{\lambda_0 \lambda_1}\|P^j f\|. \end{aligned}$$

Set $\lambda = 2\sqrt{\lambda_0 \lambda_1}$. We need to set the θ_j's such that the inequality $a=b$ will hold. In other words, we need $\lambda_0 \sum_{j=1}^{q} \theta_j^{-1} \|P^j f\|^2 = \lambda_1 \sum_{j=1}^{q} \theta_j$. For each $j = 1,\ldots,q$:

$$\begin{aligned} \lambda_0 \theta_j^{-1} \|P^j f\|^2 &= \lambda_1 \theta_j \\ \theta_j^2 &= \frac{\lambda_0}{\lambda_1} \|P^j f\|^2 \\ \Rightarrow \theta_j &= \sqrt{\frac{\lambda_0}{\lambda_1}} \|P^j f\|. \end{aligned}$$

Set each $\theta_j = \sqrt{\frac{\lambda_0}{\lambda_1}} \|P^j f\|$, then the proof of lemma follows.

We would like to emphasize an important issue about the θ's at this step. The main purpose of these θ_j's in smoothing spline ANOVA models are to scale the penalization of each component in the model. They are called smoothing parameters, and a careful selection of these parameters is crucial in order to provide a good SS-ANOVA estimate. As mentioned earlier in the introduction chapter, the selection of these parameters is a primary bottleneck in front of these methods. A good amount of literature is devoted to this issue. Some methods suffer from large dimensional grid search, while others recommend estimation of these parameters simultaneously. Especially in correlated data situations, estimation of the smoothing parameters and the covariance parameters simultaneously is recommended. One approach to overcome this issue is to use the linear mixed effects model representation which is discussed in Section 1.5.3; however, these methods have the disadvantage of curse of dimensionality. The main problem with the latter methods is the large number of random effects to be fitted which depends on how many components are included in the model.

In Correlated COSSO method, these θ_j parameters are important for both scaling the roughness penalties for different components and controlling the selection of components. One should remark that the form of (2.5) looks similar to a SS-ANOVA model with multiple smoothing parameters except that θ's are fixed rates to be estimated rather than a set of smoothing parameters. A clever algorithm will be provided in the next section to find the solution to (2.5). The algorithm estimates the set of θ parameters directly, hence no tuning will be necessary for these fixed rates. In addition, the formulation includes an extra penalty term for the set of θ parameters. This extra penalization shrinks these parameters towards zero, and hence provides sparse solution. The model sparsity follows the sparsity of these parameters, since for any j, if $\hat{\theta}_j = 0$ then the corresponding component will disappear from the model.

We also remark that λ_0 is a constant, which can be fixed at any positive value. The reason for λ_0 being included in the formulation is that it helps to apply a natural scale to the smoothing parameter (λ_1), hence stabilizes the computation. The only smoothing parameter in the Cor-

COSSO is λ_1. The methods to be used to tune this parameter will be discussed in Section 2.3.2.

2.3 Computation of Cor-COSSO

In the previous two sections of this chapter, we formalize the Correlated COSSO method by defining the objective function explicitly and its equivalent formulation. In this section we propose the iterative algorithms for the computation of Correlated COSSO.

For fixed $\theta, \tau, \lambda_0, \lambda_1$, Theorem 2.2.3 states that the minimizer of (2.5) has the form:

$$f(\mathbf{x}) = d + \sum_{i=1}^{n} c_i R_\theta(\mathbf{x}_i, \mathbf{x}),$$

where $R_\theta(\cdot,\cdot) = \sum_{j=1}^{q} \theta_j R_j(\cdot,\cdot)$ and $R_j(\cdot,\cdot)$ is the reproducing kernel of $\mathscr{H}^j$ as defined previously. With some abuse of notation, we use R_j for the matrix $\{R_j(\mathbf{x}_i, \mathbf{x}_{i'})\}_{i,i'=1}^{n}$. Let $\mathbf{c} = (c_1, ..., c_n)^{\mathrm{T}}$, $\mathbf{f}_j = \left(f_j(x_1^{(j)}), ..., f_j(x_n^{(j)}))\right)^{\mathrm{T}}$, and $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_n))^{\mathrm{T}}$. Let $\mathbf{1}$ be the vector of ones of length n. Then we have:

$$\mathbf{f}_j = \theta_j R_j \mathbf{c}, \quad \mathbf{f} = \mathbf{1}d + \sum_{j=1}^{q} \mathbf{f}_j = \mathbf{1}d + R_\theta \mathbf{c},$$

and the penalty term

$$\sum_{j=1}^{q} \theta_j^{-1} \|P^j f\|^2 = \sum_{j=1}^{q} \theta_j \mathbf{c}^{\mathrm{T}} R_j \mathbf{c} = \mathbf{c}^{\mathrm{T}} R_\theta \mathbf{c}.$$

Therefore the Correlated COSSO problem in (2.5) becomes:

$$\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{1}d-R_\theta\mathbf{c})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{1}d-R_\theta\mathbf{c})+n\lambda_0\mathbf{c}^{\mathrm{T}}R_\theta\mathbf{c}+n\lambda_1\mathbf{1}_q^{\mathrm{T}}\theta. \tag{2.6}$$

2.3.1 Cor-COSSO Algorithms with Fixed Tuning Parameters

The objective function of Cor-COSSO (2.6) can be minimized with respect to all the parameters $\mathbf{c}, d, \theta, \sigma^2$ and τ jointly. However, we propose an iterative approach by taking advantage of existing software packages. For this moment, assume the tuning parameter λ_1 is fixed.

We propose to minimize (2.6) with the two following stages:

1. **ESTIMATION** Stage: With θ fixed, solve $(d, \mathbf{c}, \tau, \sigma^2)$ using the following procedure.

   With $(\sigma^2, \tau, \lambda_0)$ fixed, we solve penalized weighted least squares problem:

   $$\min_{d,\mathbf{c},}\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{1}d-R_\theta\mathbf{c})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{1}d-R_\theta\mathbf{c})+n\lambda_0\mathbf{c}^{\mathrm{T}}R_\theta\mathbf{c}.$$

   We treat $(\sigma^2, \tau, \lambda_0)$ as variance components, and estimate them using Generalized Maximum Likelihood method. Simultaneous estimation of these parameters is done with linear mixed model connection.

   Define the solution as $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2})$.

2. **SELECTION** Stage: Let $\lambda_0^* = 2\lambda_0\sigma^2$. With $(d, \mathbf{c}, \tau, \sigma^2)$ fixed at $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2})$, the solu-

tions from ESTIMATION stage, solve the optimization problem for θ:

$$\min_{\theta} \left(\mathbf{y} - \mathbf{1}\hat{d} - R_{\theta}\hat{\mathbf{c}}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \left(\mathbf{y} - \mathbf{1}\hat{d} - R_{\theta}\hat{\mathbf{c}}\right) + n\lambda_0^{*}\hat{\mathbf{c}}^{\mathrm{T}} R_{\theta}\hat{\mathbf{c}}$$

subject to $\sum_{j=1}^{q} \theta_j \leq M, \quad \theta_j \geq 0, \quad j = 1, \ldots, q.$

Here $M \geq 0$ is the tuning parameter, which is one-to-one corresponding to λ_1. Let us take a closer look at both stages.

ESTIMATION Stage:

In the ESTIMATION stage, we fix θ, therefore the third quantity in (2.6) becomes irrelevant. Absorbing σ^2 to the smoothing parameter when σ^2 and τ are fixed, the Correlated COSSO minimization problem (2.6) becomes:

$$\min_{d,\mathbf{c}} \left(\mathbf{y} - \mathbf{1}d - R_{\theta}\mathbf{c}\right)^{\mathrm{T}} \mathbf{W}_{\tau} \left(\mathbf{y} - \mathbf{1}d - R_{\theta}\mathbf{c}\right) + n\lambda_0^{*}\mathbf{c}^{\mathrm{T}} R_{\theta}\mathbf{c}, \tag{2.7}$$

where $\lambda_0^{*} = 2\lambda_0\sigma^2$.

A closer look at this equation will reveal that this is not different from a traditional smoothing spline ANOVA formulation. Besides, since the θ_j's are fixed, we only have one smoothing parameter (λ_0^{*}) in the formulation. This will automatically reduce the multidimensional smoothing parameter selection problem to a one dimensional problem.

Now we should make an important remark about λ_0^{*}. In the original formulation of Cor-

COSSO, it is stated that this parameter can be fixed at any positive constant. On the other hand, we would like to select this positive constant properly in order to find a consistent range of λ_1 for tuning. Therefore, in practice, we recommend to adaptively choose the λ_0^* in the ESTIMATION stage. This parameter will be treated as a variance component at this stage, and will be estimated using a linear mixed model representation, which will be covered next.

We use the penalized weighted least squares approach to estimate the functional components in SS-ANOVA problem. The representer theorem of Kimeldorf and Wahba (1971) guarantees that the SS-ANOVA solution $\hat{\mathbf{f}} = \mathbf{1}\hat{d} + R_\theta \hat{\mathbf{c}}$ has a finite dimensional representation. Taking the derivatives of (2.7) with respect to d and $\mathbf{c}$ and equating them to 0 give the following equation system to find $\hat{d}$ and $\hat{\mathbf{c}}$:

$$\begin{aligned} \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau R_\theta \mathbf{c} + \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau \mathbf{1} d &= \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau \mathbf{y}, \\ \left(R_\theta^{\mathrm{T}}\mathbf{W}_\tau R_\theta + n\lambda_0^* R_\theta\right)\mathbf{c} + R_\theta^{\mathrm{T}}\mathbf{W}_\tau \mathbf{1} d &= R_\theta^{\mathrm{T}}\mathbf{W}_\tau \mathbf{y}. \end{aligned} \tag{2.8}$$

In order to estimate the variance-covariance parameters (σ^2, τ) and the smoothing parameter (λ_0^*) simultaneously, we recommend using the GML approach, which is presented in Section 1.5.2.

Consider the following linear mixed effects model:

$$\mathbf{y} = \mathbf{1}d + \mathbf{u} + \varepsilon, \tag{2.9}$$

where $\mathbf{u} \sim N\left(\mathbf{0}, \sigma^2 R_\theta/(n\lambda_0^*)\right)$ and $\varepsilon \sim N\left(\mathbf{0}, \sigma^2 \mathbf{W}_\tau^{-1}\right)$. Fixed and random effects of the model

(2.9) is estimated using Equation (3.3) of Harville (1977). The so called mixed model normal equations gives the following matrix solution:

$$\begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau\mathbf{1} & \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau R_\theta \\ R_\theta\mathbf{W}_\tau\mathbf{1} & R_\theta\mathbf{W}_\tau R_\theta + n\lambda_0^* R_\theta \end{pmatrix} \begin{pmatrix} d \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_\tau\mathbf{y} \\ R_\theta\mathbf{W}_\tau\mathbf{y} \end{pmatrix}. \tag{2.10}$$

The equation system above is identical to that in (2.8). In other words, the fixed effect estimates for $\hat{d}$ are the same, while the random effects predictions ($\hat{\mathbf{u}}$) can be used to calculate $\hat{\mathbf{c}}$; i.e., $\hat{\mathbf{c}} = R_\theta^{-1}\hat{\mathbf{u}}$. Therefore, the SS-ANOVA estimate $\hat{\mathbf{f}} = \mathbf{1}\hat{d} + R_\theta\hat{\mathbf{c}}$ is the Best Linear Unbiased Predictions (BLUPs) from the linear mixed effects model. Following the discussion on the connection between the Generalized Maximum Likelihood (GML) method and the Restricted Maximum Likelihood (REML) estimate of the mixed model representation, the covariance parameters $(\lambda_0^*, \sigma^2, \tau)$ in (2.9) are estimated with the REML method.

In summary, we use the linear mixed effects model to solve the induced smoothing spline ANOVA problem when θ_j's are fixed in Correlated COSSO model. The purpose of using the linear mixed effects model representation is to benefit from the advanced theory and the computational power of these models. For example, this representation lets us use the standard commercial statistical software (such as *SAS*, *S-Plus* etc.) to solve the minimization problem in ESTIMATION stage. We implement a SAS macro which uses the *SAS Proc Mixed* software in our numerical analysis.

Another advantage of using the mixed model representation is that the variance components (τ, σ^2 and λ_0^*) can be estimated by REstricted Maximum Likelihood (REML) method (Wang, 1998b; Opsomer, Wang, and Yang, 2001). This variance component estimation provides the main motivation for Generalized Maximum Likelihood (GML) as a selection criterion for parameters. GML is shown to have superior performance compared to Unbiased Risk (UBR), Generalized Cross Validation (GCV), L method with numerous simulation results by Wang (1998b). We will follow the recommendations from these simulation examples, and use GML for estimating λ_0^*. A more detailed explanation of smoothing parameter selection methods in correlated data situations will be discussed in Section 1.5.2.

There are n random effects to be estimated in the mixed model in the ESTIMATION stage. The important point is that, the number of random effects does not depend on the number of components in the model, which is a substantial improvement compared to the SS-ANOVA recommended in Wang (1998b). Note in the traditional SS-ANOVA formulation, there are totally nq random effects, n being the sample size and q being the number of components in the model. This number can increase quickly in a large dimensional dataset, and it also inflates the computational time. By contrast, the computation time of the Cor-COSSO is not affected as much by the number of explanatory variables. This is a desired property especially for high dimensional regression and variable selection problems. Even though the primary objective of the analysis is not variable selection, the Correlated COSSO method can be applied to solve the multi-dimensional tuning issue without estimating a large number of random effects as in smoothing spline estimation.

SELECTION Stage:

In this step, the function estimate $\hat{d}, \hat{\mathbf{c}}$ and the variance-covariance parameters are fixed at their current values, and the minimization takes place over θ parameters. Let $\lambda_0^* = 2\lambda_0\sigma^2$. When $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2})$ are fixed at the solutions of the ESTIMATION stage, the equivalent formulation to Correlated COSSO becomes:

$$\min_{\theta} \left(\mathbf{y} - \mathbf{1}\hat{d} - R_\theta \hat{\mathbf{c}}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \left(\mathbf{y} - \mathbf{1}\hat{d} - R_\theta \hat{\mathbf{c}}\right) + n\lambda_0^* \hat{\mathbf{c}}^{\mathrm{T}} R_\theta \hat{\mathbf{c}} \tag{2.11}$$

subject to $\sum_{j=1}^{q} \theta_j \leq M, \theta_j \geq 0, j = 1, \ldots, q.$

Denote $g_j = R_j\hat{\mathbf{c}}$ for $j = 1, \ldots, q$, $\mathbf{G}$ as an $n \times q$ matrix with j^{th} column being g_j, equation (2.11) can be written as:

$$\begin{aligned} &\min_{\theta} \frac{1}{2}\theta^{\mathrm{T}} \left[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{\hat{\tau}}\mathbf{G}\right] \theta - \left[\left(\mathbf{y} - \mathbf{1}\hat{d}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} - n\lambda_0^*/2\hat{\mathbf{c}}^{\mathrm{T}}\right] \mathbf{G}\theta, \\ &\qquad \text{subject to} \quad \sum_{j=1}^{q} \theta_j \leq M, \quad \theta_j \geq 0, \quad j = 1, \ldots, q. \end{aligned} \tag{2.12}$$

The objective function in minimization problem in (2.12) is quadratic in θ, and the optimization is made under linear constraints and can be rephrased as:

$$\min_{\theta} \frac{1}{2}\theta^{\mathrm{T}}\mathbf{H}\theta + \mathbf{a}^{\mathrm{T}}\theta, \quad \text{subject to} \quad \mathbf{1}^{\mathrm{T}}\theta \leq M, \quad \theta_j \geq 0, \quad j = 1, \ldots, q, \tag{2.13}$$

where, $\mathbf{H} = \mathbf{G}^{\mathrm{T}}\mathbf{W}_{\hat{\tau}}\mathbf{G}$, and $\mathbf{a} = \mathbf{G}^{\mathrm{T}}\left[(n\lambda_0^*/2)\hat{\mathbf{c}} - \mathbf{W}_{\hat{\tau}}(\mathbf{y} - \mathbf{1}\hat{d})\right]$. This is the Quadratic Programming (QP) problem, and many algorithms can provide efficient solutions. In practice, we used a *SAS Proc IML* routine *qp* to solve the SELECTION stage.

In the remaining part of this section, we provide several algorithms to compute Correlated COSSO solutions. The two stages (ESTIMATION and SELECTION) is implemented in all of these algorithms iteratively. Careful readers will notice that the smoothing parameter λ_1 is reparameterized into another smoothing parameter M. They are actually equivalent, and we will discuss this issue in Section 2.3.2.

2.3.1.1 Full Iteration Algorithm

The main idea in *Full Iteration* Cor-COSSO algorithm is to iterate between ESTIMATION and SELECTION stages until convergence. At the initial step, we will start from a very basic model, where all θ_j's are fixed at 1. The iteration will be continued until the convergence is achieved. The convergence criteria is based on the sum of squared θ_j differences in successive steps. Namely:

$$\text{Stop at iteration ''}k\text{'' if} \quad \sum_{j=1}^{q} \left| \hat{\theta}_j^{(k)} - \hat{\theta}_j^{(k-1)} \right| \quad < \quad \delta, \tag{2.14}$$

where $\delta > 0$ is a prechosen small positive number, and $\hat{\theta}_j^{(k)}$ is the value of θ_j at the k^{th} iteration. The convergence criterion can be adjusted using δ.

For a fixed M, the full iteration Cor-COSSO algorithm is following:

1. Fix $\hat{\theta}_j = 1$, for $j = 1, \ldots, q$.

2. Solve (2.7) for $\mathbf{c}, d, \tau, \sigma$ and λ_0^* using the ESTIMATION stage.

3. With fixed $\hat{\mathbf{c}}, \hat{d}, \hat{\tau}, \hat{\sigma}$ and λ_0^* in step 2, solve for $\hat{\theta}$ using (2.12).

4. Iterate between steps 2 and 3 until convergence.

In practice we have encountered some problems with the full iteration Correlated COSSO algorithm. First of all, the first iteration already provides very good results, and the difference between the iterations is observed to be slow after this first step. We believe the main reason is that, we start with a good estimate, the smoothing spline solution, which makes the first update to be a good solution. Motivated by this, one alternative algorithm, the one-step algorithm is presented in Section 2.3.1.4.

Another computational problem we empirically observed is that, the final solution is not sparse for the full iteration algorithm. Starting from the second iteration, the estimates for some θ_j's are tiny, but not exactly zero. Although the corresponding components have minimal effect on the model estimation, these components can not be completely excluded from the model. We therefore propose the following two algorithms, which can further "sparsify" the solution from the full iteration Correlated COSSO algorithm.

2.3.1.2 Full Iteration Algorithm with Truncation

We note that the parameter θ_j controls the sparsity of each component f_j in the Correlated COSSO. The minimization problem (2.6) applies a shrinkage penalty $\sum_{j=1}^{q} \theta_j$, which shrinks them towards zero. If for some j, $\hat{\theta}_j = 0$, then the corresponding component $(\hat{f}_j)$ disappears from the model. However, in simulation studies on full iteration algorithm, we noticed that some $\hat{\theta}_j$ parameters are estimated very close to zero, although not exact zeros due to numerical reasons. We recommend using truncation at the end of the full iteration Correlated COSSO algorithm. The full iteration algorithm with truncation goes through steps 1-4 of the full iteration algorithm, and uses a threshold to truncate small θ_j's to exact zeros. The steps of this algorithm are following:

1. Go through Step1 to Step 4 in full iteration Correlated COSSO algorithm.

2. If $\hat{\theta}_j < \delta_0 \quad \Rightarrow$ Set $\hat{\theta}_j = 0$.

δ_0 can be selected as any small positive number. We use $\delta_0 = 0.001$ throughout this dissertation research.

2.3.1.3 Step-Down Algorithm

An alternative approach to resolve the non-sparsity issue of the full iteration algorithm is the Step-Down algorithm. We have observed that, most of the small θ_j parameters are actually exact zeros in the earlier iterations. So at each step, we may remove the components whose corresponding θ_j is estimated as exactly 0. The full description of the algorithm is following:

1. Fix $\hat{\theta}_j = 1$, for $j = 1, \ldots, q$.

2. Solve (2.7) for $\mathbf{c}, d, \tau, \sigma$ and λ_0^* using the ESTIMATION stage.

3. With fixed $\hat{\mathbf{c}}, \hat{d}, \hat{\tau}, \hat{\sigma}$ and λ_0^* in step 2, solve for $\hat{\theta}$ using (2.12). If some $\hat{\theta}_j = 0$, then remove the corresponding component from the model.

4. Iterate between steps 2 and 3 until convergence.

In practice, both truncated full iteration and step-down algorithms offer sparse solutions, hence they are practical alternatives to the full iteration algorithm. All three algorithms mentioned iterate between ESTIMATION and SELECTION stages until convergence, hence might take a long time for the Correlated COSSO model to be estimated. The following section offers an approximate algorithm which can decrease the computation time considerably.

2.3.1.4 One-Step Update Algorithm

As mentioned above, although the convergence for the variations of full iteration algorithm can be achieved in several iteration steps, the computation time might be an issue. In simulation studies, we empirically observed that the objective function of (2.6) drops drastically in the first iteration, but after this step, the solution change is not very substantial. A similar observation was made by Lin and Zhang (2006) and Zhang and Lin (2006) in the original COSSO work as well. Based on this, we propose the following one-step update algorithm as an alternative:

1. Fix $\hat{\theta}_j = 1$, for $j = 1, \ldots, q$.

2. Solve (2.7) for $\mathbf{c}, d, \tau, \sigma$ and λ_0^* using the ESTIMATION stage.

3. With fixed $\hat{\mathbf{c}}, \hat{d}, \hat{\tau}, \hat{\sigma}$ and λ_0^* in step 2, solve for $\hat{\theta}$ using (2.12).

4. With the new $\hat{\theta}$, solve for the correlated SS-ANOVA using the ESTIMATION stage optimization.

Since the non-sparsity issue of the full iteration Correlated COSSO algorithm usually occurs after the second step, the one-step update algorithm does not have the non-sparsity issue.

The four algorithm mentioned above will be compared in a simulation study in Section 5.2.2. Based on the discussion in this simulation study, we recommend using the One-Step Update algorithm because of its advantages in computation time and sparsity, with comparably good performance with the full iteration algorithms.

All four algorithms mentioned above assume a fixed M. The smoothing parameter M controls the number of variables appearing in the final model, so it should be tuned carefully to achieve a parsimonious but sufficient model. Inside the algorithm, another parameter λ_0^* is fixed at a positive constant. However, we recommend to estimate this parameter using Restricted Maximum Likelihood estimation. This corresponds to the GML method (see Wang 1998b). Selection of parameters λ_0^* and M is discussed in the next section.

2.3.2 Smoothing Parameter Selection

The prediction performance of any nonparametric regression estimate depends heavily on the choice of smoothing parameters. As an example, the smoothing spline fit might range from an interpolation to a fully parametric fit based on the choice of the tuning parameter. In the

Correlated COSSO, the optimal choice of smoothing parameters is important to assure a good model fit and component selection performance.

In nonparametric regression literature, several tuning criteria have been proposed and widely used, including Cross Validation (CV), Generalized Cross Validation (GCV, Craven and Wahba 1979), Unbiased Risk (UBR). However, these methods break down in the presence of correlation between errors (Altman, 1990; Diggle and Hutchinson, 1989; Opsomer, Wang, and Yang, 2001; Gu and Han, 2004). An alternative L method is recommended by Diggle and Hutchinson (1989). Extensions of GCV and UBR were proposed in Wang (1998b). An extension of Generalized Maximum Likelihood (GML) criterion of Wahba (1985) to correlated data is studied by both Wang (1998b) and Opsomer, Wang, and Yang (2001). Treating the smoothing parameter as a variance component and estimating them via Restricted Maximum Likelihood (REML) from the mixed model estimation makes the GML an appealing procedure (Wang, 1998b).

Two parameters to be selected in Correlated COSSO method are λ_0^* and M. In theory, λ_0^* can be fixed at any positive constant, however, for computational purposes, we estimate this parameter in the ESTIMATION stage. The parameter M plays the most important role for both model estimation and variable selection. We recommend using a grid-search for M in the range of 0 to q, where q is the number of components in the full model.

2.3.2.1 Selection of Parameter for Model Smoothing

At the ESTIMATION stage of the Correlated COSSO algorithm, we essentially fit a SS-ANOVA model (with fixed values of θ's). In this formulation, λ_0^* is the only smoothing parameter to be selected. We recommend to treat λ_0^* as a variance component and estimate it with other variance-covariance parameters altogether (Wang, 1998b; Opsomer, Wang, and Yang, 2001).

Throughout this dissertation research, we use GML for selecting λ_0^*. The GML method is suggested by Wang (1998b) and Opsomer, Wang, and Yang (2001). One advantage of the GML method is that it treats λ_0^* as a variance component and automatically estimates it in the mixed model framework. Based on Section 1.5.2, the GML estimators of the variance components τ and λ_0^* for the ESTIMATION step (2.7) are the minimizers of:

$$M(\tau,\lambda_0^*) = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{W}_\tau (I-\mathbf{A})\,\mathbf{y}}{\left[\det^{+}\left(\mathbf{W}_\tau (I-\mathbf{A})\right)\right]^{\frac{1}{n-1}}}, \tag{2.15}$$

where $\det^{+}$ is the product of nonzero eigenvalues, and $\mathbf{A}$ is the hat or influence matrix of SS-ANOVA model defined as:

$$\mathbf{A} = I - n\lambda_0^*\mathbf{W}_\tau^{-1}Q_2\left[Q_2^{\mathrm{T}}(R_\theta + n\lambda_0^*\mathbf{W}_\tau^{-1})Q_2\right]^{-1}Q_2^{\mathrm{T}}.$$

Other tuning criteria such as GCV, UBR or L-method can also be used to tune λ_0^* in SS-ANOVA model. However, these methods require knowledge of σ^2 and τ, hence they do not estimate these parameters simultaneously with λ_0^*. Moreover, a grid search on λ_0^* is needed for the other methods. Wang (1998b) conducted a simulation study to compare these methods with GML for tuning λ_0^* and found that GML outperformed all other methods. Therefore, we follow the recommendations from Wang (1998b), and use GML method to estimate λ_0^*.

2.3.2.2 Selection of Parameter for Model Sparsity

The smoothing parameter M is fixed when we describe the computational algorithms in the previous section. M is a positive constant controlling the number of components included in the model. Therefore, the proper selection of this parameter is crucial for both model accuracy and variable selection performance. Now we propose various tuning criteria including the UBR, GCV and WMSE (Wang, 1998b), and L method from Diggle and Hutchinson (1989) to tune M. These methods will be compared with a simulation study in Section 5.2.1.

The weighted mean squared errors (WMSE) is defined as:

$$T_k = \frac{1}{n}(\hat{\mathbf{f}} - \mathbf{f})^{\mathrm{T}} \mathbf{W}_\tau^k (\hat{\mathbf{f}} - \mathbf{f}) = \frac{1}{n} \|\mathbf{W}_\tau^{k/2} (\hat{\mathbf{f}} - \mathbf{f})\|, \quad k = 0, 1, 2,$$

where k can be considered as the parameter to control the contribution of the covariance structure to model tuning. Remark that if $k = 0$, then we treat the data as independent.

One important caveat of WMSE methods is that, its calculation requires the knowledge of the regression function (f). This information is however seldom available for real data sets. In simulation studies, since we know the real data generating process, we can calculate WMSE.

This motivates us to find an unbiased estimate of T_k.

Plugging $\hat{\mathbf{f}} = \mathbf{A}\mathbf{y}$, the expected value of the WMSE can be computed as:

$$ET_k = \frac{1}{n}\mathbf{f}^{\mathrm{T}}(I-\mathbf{A}^{\mathrm{T}})\mathbf{W}_{\tau}^{k}(I-\mathbf{A})\mathbf{f} + \frac{\sigma^2}{n}\mathrm{Tr}(\mathbf{A}^{\mathrm{T}}\mathbf{W}_{\tau}^{k}\mathbf{A}\mathbf{W}_{\tau}^{-1}), \quad k = 0,1,2.$$

An unbiased estimate of ET_k is then given by:

$$U_k = \frac{1}{n}\mathbf{y}^{\mathrm{T}}(I-\mathbf{A}^{\mathrm{T}})\mathbf{W}_{\tau}^{k}(I-\mathbf{A})\mathbf{y} - \frac{\sigma^2}{n}\mathrm{Tr}(\mathbf{W}_{\tau}^{k-1}) + 2\frac{\sigma^2}{n}\mathrm{Tr}(\mathbf{W}_{\tau}^{k-1}\mathbf{A}), \quad k = 0,1,2. \qquad (2.16)$$

Here U_k's are called the Unbiased Risk (UBR) scores, and can be used as an alternative criteria to choose the smoothing parameter M.

Another popular method to tune M is the Generalized Cross Validation (GCV, Craven and Wahba 1979). In particular, define:

$$GCV_k = \frac{\frac{1}{n}\|\mathbf{W}_{\tau}^{k/2}(I-\mathbf{A})\mathbf{y}\|^2}{[\frac{1}{n}\mathrm{Tr}(\mathbf{W}_{\tau}^{k-1}(I-\mathbf{A}))]^2}, \qquad k = 0,1,2. \qquad (2.17)$$

The estimate of M which minimizes GCV_k is called the GCV estimate.

The last method for tuning M is from Diggle and Hutchinson (1989) and will be referred to as the L method. This method requires the minimization of the following L function:

$$L = n\log\left[\mathbf{y}^{\mathrm{T}}(I-\mathbf{A}^{\mathrm{T}})\mathbf{W}_{\tau}(I-\mathbf{A})\mathbf{y}\right] + \log|\mathbf{W}_{\tau}^{-1}| + \log(n)\mathrm{Tr}\mathbf{A}. \qquad (2.18)$$

As a summary, in this section we mention about four tuning criteria (GML, UBR, WMSE and *L*). Except the *L* method, all three criterion have variations which adjust for the correlation structure in the dataset. In other words, the criteria WMSE, UBR and GCV have three versions ($k = 0, 1, 2$). The WMSE criteria requires the knowledge of the regression function that generates the data, which makes this criteria unavailable for practice. Also UBR requires the knowledge of σ^2, which can be estimated using the residuals.

In order to compare the performances of these criteria, we present several simulation studies in Section 5.2.1. Our results show that *L* method works the best in most cases. We suggest using this criteria for tuning *M* in practice.

2.3.3 Complete Algorithm for Cor-COSSO

The following is the complete algorithm for the Correlated COSSO method. Users have the flexibility to select from four computation algorithms (Full Iteration w/o truncation, step-down, one-step update), and from four types of tuning criteria ($WMSE_k$, GCV_k, UBR_k and *L*) for M.

1. Fix $\hat{\theta}_j = 1$, for $j = 1, \ldots, q$.

2. Solve (2.7) for $\hat{\mathbf{c}}, \hat{d}, \hat{\tau}, \hat{\sigma}$ and λ_0^* using the ESTIMATION stage. Fix λ_0^* for the rest of the algorithm.

3. For each value in a reasonable grid of M, fix this parameter.

4. Use the algorithms above (full iteration, full iteration with truncation, step-down, one-step update) to fit Cor-COSSO model.

5. Calculate the tuning score of the selected tuning criterion.

6. Repeat steps 4 and 5 for the whole tuning grid of M.

7. Find the minimal tuning score. The corresponding solution is the Correlated COSSO estimate.

One important decision in this algorithm is how to select the range and grid of M for tuning. The estimation of λ_0^* parameter is important especially for this point. The adaptive selection of this parameter in ESTIMATION stage provides an easily definable range for M. We recommend tuning M in the range from 0 to q, where q is the number of components in the full model. We use the positive integers within this range as the grid for tuning.

APPENDIX

Proof of Lemma 2.2.1

a. Continuity: We define the convergence as convergence by the norm i.e., $f_n \rightarrow f$ means $\|f_n \rightarrow f\| \rightarrow 0$.

Let $f_n, f \in \mathscr{H}$ and $f_n \rightarrow f$ as $n \rightarrow \infty$. Then we need to show $\Big|A(f_n) - A(f)\Big| \rightarrow 0$.

Consider $L(\cdot)$;

$$\begin{aligned} \Big|L(f_n) - L(f)\Big| &= \left|\frac{1}{n}\Big(\mathbf{y} - \mathbf{f}_n\Big)^{\mathrm{T}}\mathbf{W}\Big(\mathbf{y} - \mathbf{f}_n\Big) - \frac{1}{n}\Big(\mathbf{y} - \mathbf{f}\Big)^{\mathrm{T}}\mathbf{W}\Big(\mathbf{y} - \mathbf{f}\Big)\right| \\ &= \left|\frac{1}{n}\Big(\mathbf{f} - \mathbf{f}_n\Big)^{\mathrm{T}}\mathbf{W}\Big(\mathbf{f} - \mathbf{f}_n\Big)\right| \rightarrow 0 \end{aligned}$$

as $\mathbf{f}_n \rightarrow \mathbf{f}$. Now consider $J(f)$;

$$\begin{aligned} \left|\sum_{j=1}^{q} \|P^j f_n\| - \sum_{j=1}^{q} \|P^j f\|\right| &= \left|\sum_{j=1}^{q} \Big(\|P^j f_n\| - \|P^j f\|\Big)\right| \\ &\leq \left|\sum_{j=1}^{q} \Big(\|P^j f_n - P^j f\|\Big)\right| \\ &= \left|\sum_{j=1}^{q} \Big(\|P^j (f_n - f)\|\Big)\right| \rightarrow 0 \end{aligned}$$

as $f_n \rightarrow f$.

Therefore $\Big|A(f) - A(f_n)\Big| = \Big|L(\mathbf{f}) + J(f) - L(\mathbf{f}_n) - J(f_n)\Big| \rightarrow 0$, hence $A(f)$ is continuous.

b. Convexity: $L(\mathbf{f})$ is convex because, $\frac{\partial^2 L(\mathbf{f})}{\partial \mathbf{f}^2} = \frac{2}{n}\mathbf{W}$ is positive definite. In order to prove the convexity of $A(f)$, we need to show that

$$J\Big(\alpha f + (1-\alpha)g\Big) \leq \alpha J(f) + (1-\alpha)J(g)$$

for any $0 \leq \alpha \leq 1$.

$$\begin{aligned}
\sum_{j=1}^{q} \Big\| P^j\{\alpha f + (1-\alpha)g\} \Big\| &= \sum_{j=1}^{q} \Big\| \alpha P^j(f) + (1-\alpha)P^j(g) \Big\| \\
&\leq \sum_{j=1}^{q} \Big\{ \alpha \big\| P^j f \big\| + (1-\alpha) \big\| P^j g \big\| \Big\} \\
&= \alpha \sum_{j=1}^{q} \big\| P^j f \big\| + (1-\alpha) \sum_{j=1}^{q} \big\| P^j g \big\| \\
&= \alpha J(f) + (1-\alpha)J(g).
\end{aligned}$$

Therefore, $A(f) = L(\mathbf{f}) + J(f)$ is convex, and following a and b completes the proof of Lemma 2.2.1.

Proof of Lemma 2.2.2 *a. Convexity:* Let $f, g \in \mathscr{D}$, $0 \leq \alpha \leq 1$. Show that $h = \alpha f + (1 - \alpha)g \in \mathscr{D}$.

First, lets show the convexity of $\mathscr{H}$. Let $f, g \in \mathscr{H}$, then $f = b + f_1, \quad g = \tilde{b} + g_1 \text{where} b, \tilde{b} \in [1]$, and $f_1, g_1 \in \mathscr{H}_1$. Then h can be defined as $h = c + h_1, \text{where} c = \alpha b + (1-\alpha)\tilde{b}, \text{and} h_1 = \alpha f_1 + (1-\alpha)g_1$. It is obvious that $c \in [1]$, and we only need to show that $h_1 \in \mathscr{H}_1$.

Remark that $\mathscr{H}_1 = \oplus_{j=1}^{q} \mathscr{H}^j$ where $\mathscr{H}^j$'s are Sobolev Spaces.

$$\begin{aligned}
f_1, g_1 \in \mathscr{H}_1 \quad &\Rightarrow \quad f_1 = \sum_{j=1}^{q} f_1^{(j)}, g_1 = \sum_{j=1}^{q} g_1^{(j)} \text{such that} f_1^{(j)}, g_1^{(j)} \in \mathscr{H}^j, \forall j = 1, \ldots, q \\
\Rightarrow h_1 \quad &= \quad \alpha f_1 + (1-\alpha) g_1 \\
&= \quad \alpha \sum_{j=1}^{q} f_1^{(j)} + (1-\alpha) \sum_{j=1}^{q} g_1^{(j)} \\
&= \quad \sum_{j=1}^{q} \left\{ \alpha f_1^{(j)} + (1-\alpha) g_1^{(j)} \right\} \\
&= \quad \sum_{j=1}^{q} h_1^{(j)},
\end{aligned}$$

where $h_1^{(j)} = \alpha f_1^{(j)} + (1-\alpha) g_1^{(j)}$ for $j = 1, \ldots, q$.

However, $\mathscr{H}^j$'s are linear spaces, therefore $h_1^{(j)} \in \mathscr{H}^j \quad \forall j = 1, \ldots, q$, which follows $h_1 = \sum_{j=1}^{q} h_1^{(j)} \in \mathscr{H}_1$, hence $\mathscr{H}$ is convex.

If $\tilde{l} = 0$, in order to show $\mathscr{D}$ is convex, we need to show that $J(h) \leq \varsigma$.

$$\begin{aligned}
J(h) \quad &= \quad \sum_{j=1}^{q} \left\| P^j h \right\| \\
&= \quad \sum_{j=1}^{q} \left\| P^j \{ \alpha f + (1-\alpha) g \} \right\| \\
&\leq \quad \alpha \sum_{j=1}^{q} \left\| P^j f \right\| + (1-\alpha) \sum_{j=1}^{q} \left\| P^j g \right\| \\
&= \quad \alpha J(f) + (1-\alpha) J(g) \\
&\leq \quad \alpha \varsigma + (1-\alpha) \varsigma \\
&= \quad \varsigma,
\end{aligned}$$

hence, $\mathscr{D}$ is convex.

To show that $\mathscr{D}$ is convex in case of $\tilde{l} > 0$, we also need to show $|c| \leq \frac{1}{\tilde{l}}K$, where $K = \frac{\varsigma^{1/2}}{\sqrt{\tilde{d}}} + (1+a)\varsigma$.

$$\begin{aligned}
\tilde{l}\big|c\big| &= \tilde{l}\Big|\alpha b + (1-\alpha)\tilde{b}\Big| \\
&\leq \alpha\tilde{l}\big|b\big| + (1-\alpha)\tilde{l}\big|\tilde{b}\big| \\
&\leq \alpha K + (1-\alpha)K \\
&= K.
\end{aligned}$$

While passing from second to third line, we used the fact that since $f, g \in \mathscr{D}$, by the definition of the set, $|b| \leq \frac{1}{\tilde{l}}K$ and $|\tilde{b}| \leq \frac{1}{\tilde{l}}K$. Hence, $\mathscr{D}$ is convex.

b. Boundedness: Show that $\forall f \in \mathscr{D}, \exists M$ such that $\big\|f\big\| \leq M$.

We know $\big\|f\big\| \leq J(f) \leq \varsigma \quad \forall f \in \mathscr{D}$. Then let $M = \varsigma + 1 \Rightarrow \big\|f\big\| \leq M \quad \forall f \in \mathscr{D}$, hence, $\mathscr{D}$ is bounded.

c. Closedness: Let $\{f^{(m)}\} \in \mathscr{D}$, and $\{f^{(m)}\} \to f$. We need to show that $f \in \mathscr{D}$ to prove that $\mathscr{D}$ is closed.

From the definition of $\mathscr{H}$:

$\forall m; f^{(m)} = b^{(m)} + f_1^{(m)}$ such that $b^{(m)} \in [1]$ and $f_1^{(m)} \in \mathscr{H}_1$.

$f^{(m)} \to f$, so $f = b + f_1$ such that $b^{(m)} \to b$ and $f_1^{(m)} \to f_1$. It is obvious that $b \in [1]$. Now we need to show that $f_1 \in \mathscr{H}_1$.

Consider $f_1^{(m)}$:

$f_1^{(m)} = f_1^{(m,1)} + f_1^{(m,2)} + \ldots + f_1^{(m,q)}$ such that $f_1^{(m,j)} \in \mathscr{H}^j$ for $j = 1, \ldots, q$.

$f^{(m)} \to f \Rightarrow f_1^{(m,j)} \to f_1^j$ as $m \to \infty$ for $j = 1, \ldots, q$.

However, $\mathscr{H}^j$'s are Sobolev spaces (hence closed). Therefore $f_1^j \in \mathscr{H}^j$ for $j = 1, \ldots, q$.

$\Rightarrow f_1 = \sum_{j=1}^q f_1^j$ where $\forall j = 1, \ldots, q, f_1^j \in \mathscr{H}^j$, i.e., $f_1 \in \mathscr{H}_1$.

Therefore, by the definition of $\mathscr{H}$, $f = b + f_1 \in \mathscr{H}$, hence $\mathscr{H}$ is closed.

If $\tilde{l} = 0$, in order to show $\mathscr{D}$ is closed, it is enough to show that $J(f) < \varsigma$. We have already shown that $J(f)$ is continuous. By definition of continuity:

$f^{(m)} \to f \Rightarrow J\big(f^{(m)}\big) \to J(f)$ as $m \to \infty$,

which follows:

$\forall \varepsilon > 0, \exists n$ such that $\forall i \geq n, \Big| J\big(f^{(i)}\big) - J\big(f\big) \Big| < \varepsilon$.

Assume $J(f) > \varsigma$ and let $\varepsilon = \frac{1}{2}\big(J(f) - \varsigma\big)$.

$$\begin{aligned}\Big| J\big(f\big) - J\big(f^{(i)}\big) \Big| &= J\big(f\big) - J\big(f^{(i)}\big) \\ &\geq J\big(f\big) - \varsigma \\ &= 2\varepsilon,\end{aligned}$$

which is a contradiction. Therefore $J(f) \leq \varsigma$, hence $\mathscr{D}$ is closed.

To show that $\mathscr{D}$ is closed in case of $\tilde{l} > 0$, in addition, we need to show that $\big|b\big| \leq \frac{1}{\tilde{l}}K$. Similarly, $b^{(m)} \to b \Rightarrow \forall \varepsilon > 0, \exists n$ such that $\forall i \geq n, \big|b^{(m)} - b\big| < \varepsilon$.

Assume $\tilde{l}\big|b\big| > K$ and let $\varepsilon = \frac{1}{2}\Big(\tilde{l}\big|b\big| - K\Big)$.

$$\begin{aligned}
\tilde{l}\Big|b - b^{(i)}\Big| &= \tilde{l}\Big(\big|b\big| - \big|b^{(i)}\big|\Big) \\
&= \tilde{l}\big|b\big| - \tilde{l}\big|b^{(i)}\big| \\
&\geq \tilde{l}\big|b\big| - K \\
&= 2\varepsilon,
\end{aligned}$$

which is a contradiction. Therefore $\big|b\big| \leq \frac{1}{\tilde{l}}K$, hence, $\mathscr{D}$ is closed.

Following a, b and c, $\mathscr{D}$ is closed, bounded and convex sets.

CHAPTER 3

ADAPTIVE CORRELATED COSSO METHOD

3.1 Introduction

The Correlated COSSO method of Chapter 2 is shown to achieve sparse solution; however, it penalizes each component equally. Hence, in order to achieve sparse solution, the important components suffer from large bias. Therefore, the method results in oversmooth estimates for the nonzero components.

In this chapter, we present an improvement on the Correlated COSSO method by introducing a set of adaptive weights in the penalty term. Adaptive Correlated COSSO applies different scales of penalization to different components: unimportant variables receive larger penalties than important variables, and are therefore more likely to be removed from the final model. The amount of penalization is controlled by adaptively chosen weights. The name *adaptive* implies that the weights will be chosen by the data itself.

The motivation behind the weighted penalization is the Adaptive LASSO idea of Zou (2006) and Zhang and Lu (2007). This idea is also implemented successfully into the original COSSO by Storlie et al. (2007). In order to explain the implications of adaptive weights, we first present a short review of the Adaptive LASSO and Adaptive COSSO methods in Section 3.2.

We then present our formulation for the Adaptive Correlated COSSO in Section 3.3. One important question arising in the Adaptive Correlated COSSO is how to define the weights properly, so that heavier penalties are imposed on uninformative functional components while smaller penalties on the informative ones. These different penalty scales can be achieved by using a set of *fixed* positive weights, such that larger weights are associated with heavier penalties, and weights closer to 0 are associated with smaller penalties. A smaller penalty is equivalent to less shrinkage to the functional components, and a larger penalty forces the function component to shrink towards zero faster. Therefore, if the weights are selected effectively, the final model will exclude unimportant components more forcefully and produce smaller bias on the important components.

In order to define appropriate weights, a good initial estimate for f is needed. We propose using the traditional smoothing spline ANOVA fit for correlated data (1.23), where $\theta_j = 1$ for each $j = 1, \ldots, q$, and λ_0 is estimated using GML approach. The weights are selected as the inverse of the L_2 norms of corresponding components, and details are given in Section 3.3.1.

An important issue, the existence of the Adaptive Cor-COSSO solution, will be discussed in Section 3.3.2. We discuss the similarities of the Adaptive Cor-COSSO and the Correlated COSSO, and remark that this resemblance leads to the same algorithms to find the solution for Adaptive Cor-COSSO method. An equivalent formulation of Adaptive Cor-COSSO will be provided in Section 3.3.3.

The computation of Adaptive Cor-COSSO is quite parallel to that of Correlated COSSO, hence also having two main stages: ESTIMATION and SELECTION. Various algorithms are given in Section 3.4.

The smoothing parameter selection criteria are exactly the same as the Correlated COSSO criteria, hence we will not repeat them in this chapter. Extra simulation studies which are not presented in this dissertation, show that the *L* method works very well for Adaptive Correlated COSSO as well. Hence, we recommend using this smoothing parameter selection criteria.

3.2 Reviews on Adaptive LASSO and Adaptive COSSO

In this section, we shortly review some recent adaptive methods for model selection in the literature, including the Adaptive LASSO (Zou, 2006) and the Adaptive (Original) COSSO (Storlie et al., 2007).

3.2.1 LASSO and Adaptive LASSO

Recently, the shrinkage methods which simultaneously estimate model and perform variable selection are becoming popular in the literature. One of the earliest, and arguably the most famous shrinkage method for linear models is the Least Absolute Shrinkage and Selection

Operator (LASSO, Tibshirani 1996). The method is defined as:

$$\hat{\beta}_{LASSO} = \text{argmin}_{\beta} \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{3.1}$$

where λ is a positive regularization parameter. The method applies shrinkage to the least squares estimates, and λ controls the level of shrinkage. For smaller λ, the shrinkage effect is less, therefore, the estimates are close to ordinary least squares estimates. On the other hand, if λ is increased to a large enough number, then $\hat{\beta}$ estimates are shrunk more towards zeros and some variables are eventually excluded from the model.

Using the regularization parameter λ, the LASSO method applies a continuous shrinkage to the least squares estimates. This continuous shrinkage results in a much more stable estimate of β, and this is the main advantage of LASSO and LASSO type methods. Every coefficient in the model is shrunk to zero at some degree. Some components are shrunk to exactly zero and hence excluded from the model. The remaining coefficients are still nonzero, yet smaller (in absolute value) than the original least squares estimates. However, shrinkage of the important variables causes bias, and may seriously affect the large coefficients. Also, the LASSO method does not possess the desirable oracle properties (Fan and Li, 2001).

The main reason for the LASSO method to produce substantial bias in the estimates is that, the method applies the same amount of shrinkage to all variables. Zou (2006) proposed the Adaptive LASSO method, which uses adaptive weights to apply different scales of penalization to each coefficient. In particular, we have

$$\hat{\beta}_{ALASSO} = \text{argmin}_{\beta} \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|, \tag{3.2}$$

where $\mathbf{w} = (w_1, \ldots, w_p)^{\mathrm{T}} \geq \mathbf{0}$ is a weight vector. Higher weights correspond to heavier penalties, and hence more shrinkage. Naturally the weights should be chosen inversely proportional to the importance of the coefficients.

Zou (2006) shows that, if the weights are chosen properly, the formulation above provides good properties in terms of both variable selection and prediction accuracy. In particular, Zou (2006) recommends using $\hat{w}_j = 1/|\hat{\beta}_j|^{\gamma}$, where $\hat{\beta}_j$ is an initial root-n-consistent estimate of β_j. The Adaptive LASSO name comes from the fact that the weights are *adaptively* chosen from the data. It is recommended to choose initial $\hat{\beta}_j$ using ordinary least squares estimates unless there is a concern of collinearity in the data. If collinearity is expected, a more stable method such as ridge regression can be used to estimate the initial weights.

Zou (2006) also shows the oracle properties of the Adaptive LASSO estimate. In other words, the Adaptive LASSO performs as well as if the true underlying model were given in advance.

3.2.2 Adaptive COSSO Method

Component Selection and Smoothing Operator (Lin and Zhang, 2006) can be considered as the generalization of LASSO in the smoothing spline ANOVA framework. It performs component selection and model fitting via continuous shrinkage of the functional components in the SS-ANOVA model. As described previously, the COSSO penalizes the sum of norms of the functional components and achieves a sparse model by shrinking some of the components to zero functions. The number of components and their penalization is controlled by a smoothing parameter λ.

The style that COSSO works is parallel to LASSO. Therefore, the COSSO method may suffer from too much shrinkage of nonzero functions as well. The main reason behind this drawback is the fact that it forces the components to be equally penalized by its nature.

Storlie et al. (2007) proposes the Adaptive COSSO method, which aims to alleviate the tendency of oversmoothing in the COSSO using adaptive weights. The extension of COSSO to Adaptive COSSO is very similar to Zou (2006)'s Adaptive LASSO idea. The Adaptive COSSO procedure proposes to find $f \in \mathscr{H}$ to minimize:

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(x_i)]^2 + \lambda\sum_{j=1}^{q} w_j\|P^j f\|, \tag{3.3}$$

where w_j's are the *fixed* adaptive weights, which are estimated from an initial estimate of functional components, and λ is the smoothing parameter.

The selection of adaptive weights is very important and should be based on a consistent estimate. Storlie et al. (2007) recommends using a smoothing spline ANOVA, with the smoothing parameter chosen by Generalize Cross Validation (Craven and Wahba, 1979), to find the initial estimate $\hat{f}$. The weights are inversely proportional to the quantity of importance of each component. Storlie et al. (2007) showed that the Adaptive COSSO outperforms the original COSSO in various simulation studies. In the same work, they also prove that the Adaptive COSSO method possesses nonparametric oracle properties as opposed to the original COSSO method.

3.3 Adaptive Correlated COSSO Method

Consider the p-variate regression problem of (2.1) with the additive correlated errors in (2.2). In the Adaptive Correlated COSSO method, we consider the following penalized weighted least squares problem. With σ^2, τ, λ fixed, the Adaptive Correlated COSSO estimate $f \in \mathscr{H}$ minimizes:

$$\min_{f \in \mathscr{H}} \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathsf{T}}\mathbf{W}_{\tau}(\mathbf{y}-\mathbf{f}) + n\lambda \sum_{j=1}^{q} w_j \|P^j f\|, \tag{3.4}$$

where $w_j > 0$'s are the *fixed* adaptive weights that are chosen using an initial estimate of f. We call this initial estimate as $\tilde{f}$. λ is a smoothing parameter, and $\mathbf{W}_{\tau}$ is the error correlation matrix, where τ represents the correlation parameters.

Careful reader should note that the w_j's are not tuning parameters, rather they are weights to be estimated from the data. They are constructed using the initial estimate $\tilde{f}$, and hence they will not be changed during the Adaptive Correlated COSSO algorithm. The only smoothing parameter in the Adaptive Cor-COSSO formulation is λ. As discussed in the Correlated COSSO chapter, having only one smoothing parameter is an important advantage of the Adaptive Cor-COSSO method. This property of the Adaptive Cor-COSSO prevents the computational burden from multi-dimensional tuning.

One important issue is to select the initial function estimate $\tilde{f}$. Any consistent estimate for f can be used as the initial estimate. We recommend using a traditional correlated SS-ANOVA model of (1.23) with $\theta_j = 1$ for all j, and λ_0 is chosen by GML. Alternatively, the Correlated COSSO estimate of f could be used. We will discuss how to choose weights from the data, or namely the initial estimate $\tilde{f}$ in the following section.

3.3.1 Adaptive Choices of Weights

We propose to choose the weights for the Adaptive Correlated COSSO formulation using the data at hand. Ideally, smaller weights should be given to prominent components to penalize them less, while larger weights to unimportant components. Therefore, the weights should be selected inversely proportional to a measure of importance for each component. Denote the initial estimates of f by $\tilde{f}$. We use the L_2 norm of $P^j \tilde{f}$ to measure the importance of the components in the initial estimate, which was also suggested by Storlie et al. (2007) as:

$$w_j = \|P^j \hat{f}\|_{L_2}^{-\gamma},$$

where $\|.\|_{L_2}$ represents the L_2 norm of the functional component. Here $\gamma > 0$ can be regarded as a second tuning parameter, which can be adaptively chosen as well. We use $\gamma = 1$ in our examples and simulations.

3.3.2 Existence of Adaptive Cor-COSSO Solution

The existence of the Adaptive Correlated COSSO estimate is guaranteed by the following theorem.

Theorem 3.3.1 *Let $\mathscr{H}$ be an RKHS of functions over an input space $\mathscr{T}$. Assume that $\mathscr{H}$ can be decomposed as;*

$$\mathscr{H} = [1] \oplus \mathscr{H}_1 \quad \textit{with} \quad \mathscr{H}_1 = \oplus_{j=1}^{q} \mathscr{H}^j.$$

Then there exists a minimizer of (3.4) *in $\mathscr{H}$.*

The proof of the existence theorem for Adaptive Correlated COSSO is given here. The structure of this proof is also important because it reveals the connection between the Adaptive Cor-COSSO and Cor-COSSO methods. In other words, the proof actually shows that Adaptive Cor-COSSO can be considered as Cor-COSSO with an adaptive reproducing kernel Hilbert space.

Proof of Theorem 3.3.1

Define the $\tilde{\mathscr{H}}$ as the RKHS with reproducing kernel:

$$R_{\tilde{\mathscr{H}}} = 1 + \sum_{j=1}^{q} w_j^{-2} R_j(s,t),$$

where w_j's are the adaptive weights, and $R_j(\cdot,\cdot)$ is the reproducing kernel of $\mathscr{H}^j$. Note that the norm of $\tilde{\mathscr{H}}$ is:

$$\|f\|^2_{\tilde{\mathscr{H}}} = \|P^0 f\|^2_{\mathscr{H}} + \sum_{j=1}^{q} w_j^2 \|P^j f\|^2_{\mathscr{H}},$$

where the $P^0 f$ is the projection of f onto $\{1\}$, and $P^j f$'s are the projections on $\mathscr{H}^j$ spaces defined earlier. In the RKHS space $\tilde{\mathscr{H}}$ with the norm above, when we define the Cor-COSSO as in (2.4), this results in the Adaptive Correlated COSSO minimization problem. Therefore, the existence theorem of the previous chapter (Theorem 2.2.1) can be used to prove that the Adaptive Correlated COSSO estimate of (3.4) exists.

With the previous theorem, the existence of the Adaptive Cor-COSSO solution is guaranteed. The following theorem can also be proven using a parallel logic. Therefore, we skipped this proof, yet for completeness, we state the finite dimensional representation theorem.

Theorem 3.3.2 *For any fixed τ, λ and w_j for $j = 1, \dots, q$, let the minimizer of (3.4) be $\hat{f} = \hat{d} + \sum_{j=1}^{q} \hat{f}_j$, with $\hat{f}_j \in \mathscr{H}^j$. Then $\hat{f}_j \in span\{R_j(\mathbf{x}_i, \cdot), i = 1, ..., n\}$, where $R_j(\cdot,\cdot)$ is the reproducing kernel of $\mathscr{H}^j$.*

3.3.3 An Equivalent Formulation

The proof of Theorem 3.3.1 is constructive in the sense that it reveals the equivalence of an Adaptive Correlated COSSO in (3.4) to the Correlated COSSO in (2.4) with an adaptive RKHS. The computation of the methods goes parallel as well. In other words, we will follow a similar strategy to find a solution for Adaptive Cor-COSSO method.

In this part, we will introduce an equivalent formulation of (3.4) that leads naturally to an iterative algorithm. The main purpose of this equivalent formulation is to facilitate the computation of the Adaptive Cor-COSSO estimate. We will use this equivalent formulation to solve the Adaptive Correlated COSSO.

Define $\theta = (\theta_1, ..., \theta_q)^{\mathrm{T}}$ and let $\mathbf{0}$ be the vector of zeros. For fixed τ, σ^2 and w_j's, consider:

$$\min_{f \in \mathscr{H}, \theta} \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda_0 \sum_{j=1}^{p} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2 + n\lambda_1 \sum_{j=1}^{p} w_j^{\nu} \theta_j, \tag{3.5}$$

where $\forall j, \theta_j \geq 0$, $0 \leq \nu \leq 2$, $\lambda_0 > 0$ is a constant and $\lambda_1 > 0$ is a smoothing parameter. The following lemma shows that the two formulations for Adaptive Correlated COSSO have equivalent solutions.

LEMMA 3.3.1 *Set* $\lambda = 2\sqrt{\lambda_0 \lambda_1}$. *If* $\hat{f}$ *minimizes* (3.4), *set* $\hat{\theta}_j = \sqrt{\frac{\lambda_0}{\lambda_1}} w_j^{1-\nu} \|P^j \hat{f}\|$, *and then* $(\hat{\theta}, \hat{f})$ *minimizes* (3.5). *On the other hand, if* $(\hat{\theta}, \hat{f})$ *minimizes* (3.5), *then* $\hat{f}$ *minimizes* (3.4). *Therefore, solving* (3.4) *and solving* (3.5) *are equivalent.*

Proof of Lemma 3.3.1 Denote the functional in (3.4) as

$$D(f) = \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda \sum_{j=1}^{q} w_j \|P^j f\|,$$

and the functional in (3.5) as

$$B(f,\theta) = \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{f}) + n\lambda_0 \sum_{j=1}^{q} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2 + n\lambda_1 \sum_{j=1}^{q} w_j^{\nu} \theta_j.$$

We need to show that for any $0 \leq \nu \leq 2$:

$$\lambda \sum_{j=1}^{q} w_j \|P^j f\| = \lambda_0 \sum_{j=1}^{q} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2 + \lambda_1 \sum_{j=1}^{q} w_j^{\nu} \theta_j$$

to complete the proof.

We will use the inequality $a+b \geq 2\sqrt{ab}$ for $a,b \geq 0$, and equality holds if and only if $a=b$ in this proof. Hence, in our case, set $a = \lambda_0 \sum_{j=1}^{q} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2$, and $b = \lambda_1 \sum_{j=1}^{q} w_j^{\nu} \theta_j$. Then, $a+b = \lambda \sum_{j=1}^{q} w_j \|P^j f\| = 2\sqrt{\lambda_0 \sum_{j=1}^{q} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2}\sqrt{\lambda_1 \sum_{j=1}^{q} w_j^{\nu} \theta_j}$.

For each $j = 1, \ldots, q$:

$$\begin{aligned} w_j \|P^j f\| \lambda &= 2\sqrt{\lambda_0 \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2}\sqrt{\lambda_1 w_j^{\nu} \theta_j} \\ &= 2\sqrt{\lambda_0 \lambda_1 \theta_j^{-1} \theta_j w_j^{2-\nu} w_j^{\nu} \|P^j f\|^2} \\ &= w_j \|P^j f\| 2\sqrt{\lambda_0 \lambda_1}. \end{aligned}$$

Set $\lambda = 2\sqrt{\lambda_0 \lambda_1}$. We need to set the θ_j's such that the inequality $a = b$ will hold. In other words, we need $\lambda_0 \sum_{j=1}^{q} \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2 = \lambda_1 \sum_{j=1}^{q} w_j^{\nu} \theta_j$. For each $j = 1, \ldots, q$:

$$\begin{aligned} \lambda_0 \theta_j^{-1} w_j^{2-\nu} \|P^j f\|^2 &= \lambda_1 w_j^{\nu} \theta_j \\ \theta_j^2 &= \frac{\lambda_0}{\lambda_1} w_j^{2(1-\nu)} \|P^j f\|^2 \\ \Rightarrow \theta_j &= \sqrt{\frac{\lambda_0}{\lambda_1}} w_j^{1-\nu} \|P^j f\|. \end{aligned}$$

Set each $\theta_j = \sqrt{\frac{\lambda_0}{\lambda_1}} w_j^{1-\nu} \|P^j f\|$, then the proof of lemma follows.

We would like to clarify the distinction between θ_j's and w_j's. In Adaptive Correlated COSSO, parameter θ_j's have the same roles with those in Correlated COSSO. They are used for scaling the roughness penalties for different components. They are related with the component selection, since $\hat{\theta}_j = 0$ implies that the corresponding component $\hat{f}_j$ will be excluded from the model. These parameters are estimated by the Adaptive Cor-COSSO method.

On the other hand, the w_j's are fixed weights. They are not involved in the Adaptive Correlated COSSO estimation. They are kept untouched during any algorithm. In other words, prior to Adaptive Cor-COSSO algorithm, we use an initial fit to compute w_j's. As opposed to θ_j's, they do not have a direct effect on which component will be selected in the model.

Careful readers should remark that the parameter ν is not specified in the equivalent formulation proof. Hence for any constant satisfying $0 \leq \nu \leq 2$, the equivalent form can be used.

3.4 Computation of Adaptive Cor-COSSO

We already show the connection between Adaptive Correlated COSSO and Correlated COSSO. So, the algorithms we propose in this section will be parallel to the ones in Cor-COSSO method. In the previous section it is shown that (3.5) can be used to find a solution to Adaptive Cor-COSSO for any $\nu \in [0,2]$. For simplicity, we will use $\nu = 0$. In this case, the Adaptive

Cor-COSSO becomes a Cor-COSSO with an adaptive RKHS, which is stated in the proof of Theorem 3.3.1.

Let $\mathbf{w} = (w_1, \ldots, w_q)^{\mathrm{T}}$. For fixed $\theta, \tau, \sigma^2, \lambda_0, \lambda_1$, and $\mathbf{w}$, Theorem 3.3.2 states that the minimizer of (3.4) has the form:

$$f(\mathbf{x}) = d + \sum_{i=1}^{n} c_i K_{\mathbf{w},\theta}(\mathbf{x}_i, \mathbf{x}),$$

where $K_{\mathbf{w},\theta}(\cdot,\cdot) = \sum_{j=1}^{q} \theta_j w_j^{-2} R_j(\cdot,\cdot)$ and $R_j(\cdot,\cdot)$ is the reproducing kernel of $\mathscr{H}^j$ as defined previously. With some abuse of notation, we use R_j for the matrix $\{R_j(\mathbf{x}_i, \mathbf{x}_{i'})\}_{i,i'=1}^{n}$. Let $\mathbf{c} = (c_1, ..., c_n)^{\mathrm{T}}$, $\mathbf{f}_j = \left(f_j(x_1^{(j)}), ..., f_j(x_n^{(j)}))\right)^{\mathrm{T}}$, and $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_n))^{\mathrm{T}}$. Let $\mathbf{1}$ be the vector of ones of length n. Then we have

$$\mathbf{f}_j = \theta_j w_j^{-2} R_j \mathbf{c}, \quad \mathbf{f} = \mathbf{1}d + \sum_{j=1}^{q} \mathbf{f}_j = \mathbf{1}d + K_{\mathbf{w},\theta}\mathbf{c},$$

and the penalty term

$$\sum_{j=1}^{q} \theta_j^{-1} w_j^2 \|P^j f\|^2 = \sum_{j=1}^{q} \theta_j w_j^{-2} \mathbf{c}^{\mathrm{T}} R_j \mathbf{c} = \mathbf{c}^{\mathrm{T}} K_{\mathbf{w},\theta}\mathbf{c}.$$

Therefore (3.5) becomes:

$$\min_{d,\mathbf{c},\theta \geq 0} \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{1}d - K_{\mathbf{w},\theta}\mathbf{c})^{\mathrm{T}} \mathbf{W}_\tau (\mathbf{y} - \mathbf{1}d - K_{\mathbf{w},\theta}\mathbf{c}) + n\lambda_0 \mathbf{c}^{\mathrm{T}} K_{\mathbf{w},\theta}\mathbf{c} + n\lambda_1 \mathbf{1}_q^{\mathrm{T}} \theta. \tag{3.6}$$

3.4.1 Adaptive Cor-COSSO Algorithms with Fixed Tuning Parameters

Similar to Cor-COSSO, we recommend minimizing (3.6) by iterating between ESTIMATION and SELECTION stages, which will be redefined for Adaptive Correlated COSSO method. Most of the procedure is quite similar to Cor-COSSO algorithm, so we will not cover these two stages in details. We will state only the parts which are different from the Cor-COSSO method.

1. **ESTIMATION** Stage: With θ fixed, the third quantity in (3.6) disappears. With fixed $\tau, \sigma^2, \lambda_0$ and absorbing σ^2 into the smoothing parameter, (3.6) becomes:

   $$\min_{d,\mathbf{c}} \left(\mathbf{y} - \mathbf{1}d - K_{\mathbf{w},\theta}\mathbf{c}\right)^{\mathrm{T}} \mathbf{W}_\tau \left(\mathbf{y} - \mathbf{1}d - K_{\mathbf{w},\theta}\mathbf{c}\right) + n\lambda_0^* \mathbf{c}^{\mathrm{T}} K_{\mathbf{w},\theta}\mathbf{c}. \tag{3.7}$$

   where $\lambda_0^* = 2\sigma^2\lambda_0$ is the new form of the smoothing parameter. This is the classical smoothing spline ANOVA problem with correlated data (see Wang 1998b). We treat $\tau, \sigma^2, \lambda_0$ as variance components, and these parameters are simultaneously estimated by the Generalized Maximum Likelihood (GML) method.

   As it can be observed quickly, the only difference of this stage from the ESTIMATION stage of Cor-COSSO is the reproducing kernel. Correspondingly, we use the $K_{\mathbf{w},\theta}$ instead of R_θ as the RK of the adaptive space.

   We use the penalized weighted least squares approach to estimate the functional components in the SS-ANOVA problem above. The representer theorem of Kimeldorf and

Wahba (1971) guarantees that the SS-ANOVA solution $\hat{\mathbf{f}} = \mathbf{1}\hat{d} + K_{\mathbf{w},\theta}\hat{\mathbf{c}}$ has a finite dimensional representation. Taking the derivatives of (3.7) with respect to d and $\mathbf{c}$ and equating them to 0 give the following equation system to find $\hat{d}$ and $\hat{\mathbf{c}}$:

$$\begin{aligned} \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}K_{\mathbf{w},\theta}\mathbf{c} + \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{1}d &= \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{y}, \\ \left(K_{\mathbf{w},\theta}^{\mathrm{T}}\mathbf{W}_{\tau}K_{\mathbf{w},\theta} + n\lambda_0^{*}K_{\mathbf{w},\theta}\right)\mathbf{c} + K_{\mathbf{w},\theta}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{1}d &= K_{\mathbf{w},\theta}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{y}. \end{aligned} \tag{3.8}$$

We also would like to estimate the variance-covariance parameters (σ^2, τ) and the smoothing parameter (λ_0^{*}) simultaneously. We recommend using the GML approach, which is presented in Section 1.5.2. For Adaptive Correlated COSSO algorithm, ESTIMATION stage, we also use the linear mixed model representation.

Consider the following linear mixed effects model:

$$\mathbf{y} = \mathbf{1}d + \mathbf{u} + \varepsilon, \tag{3.9}$$

where $\mathbf{u} \sim N\left(\mathbf{0}, \sigma^2 K_{\mathbf{w},\theta}/(n\lambda_0^{*})\right)$ and $\varepsilon \sim N\left(\mathbf{0}, \sigma^2\mathbf{W}_{\tau}^{-1}\right)$. The estimation of fixed and random effects of model (3.9) using Equation (3.3) of Harville (1977) gives the following matrix solution:

$$\begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{1} & \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}K_{\mathbf{w},\theta} \\ K_{\mathbf{w},\theta}\mathbf{W}_{\tau}\mathbf{1} & K_{\mathbf{w},\theta}\mathbf{W}_{\tau}K_{\mathbf{w},\theta} + n\lambda_0^{*}K_{\mathbf{w},\theta} \end{pmatrix} \begin{pmatrix} d \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{W}_{\tau}\mathbf{y} \\ K_{\mathbf{w},\theta}\mathbf{W}_{\tau}\mathbf{y} \end{pmatrix}. \tag{3.10}$$

The equation system above is identical to that in (3.8). In other words, the fixed effect estimates for $\hat{d}$ are the same, while the random effect predictions ($\hat{\mathbf{u}}$) can be used to calculate $\hat{\mathbf{c}}$, i.e., $\hat{\mathbf{c}} = R_\theta^{-1}\hat{\mathbf{u}}$. Therefore, the SS-ANOVA estimate $\hat{\mathbf{f}} = \mathbf{1}\hat{d} + K_{\mathbf{w},\theta}\hat{\mathbf{c}}$ is the Best Linear Unbiased Prediction (BLUP) estimate of the linear mixed effects model. Following the discussion on the connection between the Generalized Maximum Likelihood (GML) method and the Restricted Maximum Likelihood (REML) estimate of the linear mixed model representation, the variance components of $\sigma^2, \tau, \lambda_0^*$ are estimated by REML. Careful readers should remark that λ_0^* is treated as a variance component in this representation, and is estimated with REML method along with the other variance components σ^2 and τ.

As it can be seen above, the ESTIMATION stage of Adaptive Cor-COSSO method is quite similar to the ESTIMATION stage of Cor-COSSO. The only difference is the reproducing kernel matrix R_θ is replaced by $K_{\mathbf{w},\theta}$ in every step of the computation. The Adaptive Cor-COSSO method enjoys the advantages of the linear mixed effects model connection such as the powerful theory and computational tools of mixed models.

Define the solution from the ESTIMATION stage as $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2})$.

2. **SELECTION** Stage: With $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2}, \lambda_0^*)$ fixed at the solutions of the ESTIMATION stage, solve the optimization problem for θ:

$$\min_\theta \left(\mathbf{y} - \mathbf{1}\hat{d} - K_{\mathbf{w},\theta}\hat{\mathbf{c}}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \left(\mathbf{y} - \mathbf{1}\hat{d} - K_{\mathbf{w},\theta}\hat{\mathbf{c}}\right) + n\lambda_0^* \hat{\mathbf{c}}^{\mathrm{T}} K_{\mathbf{w},\theta}\hat{\mathbf{c}} \qquad (3.11)$$

subject to $\sum_{j=1}^{q} \theta_j \leq M, \theta_j \geq 0, j = 1, \ldots, q$.

Denote $g_j = w_j^{-2} R_j \hat{\mathbf{c}}$ for $j = 1, \ldots, q$, $\mathbf{G}$ as an $n \times q$ matrix with j^{th} column being g_j. Following the discussion in Cor-COSSO chapter, the equation (3.11) can be written as a quadratic programming problem;

$$\min_{\theta} \frac{1}{2} \theta^{\mathrm{T}} \mathbf{H} \theta + \mathbf{a}^{\mathrm{T}} \theta, \quad \text{subject to} \quad \mathbf{1}^{\mathrm{T}} \theta \leq M, \quad \theta_j \geq 0 \quad j = 1, \ldots, q \tag{3.12}$$

where, $\mathbf{H} = \mathbf{G}^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \mathbf{G}$, and $\mathbf{a} = \mathbf{G}^{\mathrm{T}} \left[(n\lambda_0^*/2) \hat{\mathbf{c}} - \mathbf{W}_{\hat{\tau}} (\mathbf{y} - \mathbf{1}\hat{d}) \right]$. Reader should remark that the only difference in between (3.12) of A-Cor-COSSO and (2.13) of Cor-COSSO is the definition of $\mathbf{G}$.

In the remaining part of this chapter, we will present various algorithms for the Adaptive Correlated COSSO method. All algorithms will use the ESTIMATION and SELECTION stages as building blocks. We provide four algorithms in the previous chapter for solving the Correlated COSSO problem. All four algorithms can be readily applicable to the Adaptive Correlated COSSO problem. Therefore, we will not cover these algorithms in this chapter.

Just to give a reminder about these four algorithms, the full iteration algorithm iterates in between the ESTIMATION and SELECTION stages until convergence. However, we observe some of θ_j's provide very small numbers instead of zeros. Therefore, the full iteration with truncation algorithm is displayed, which truncates small θ_j values to exact zeros to achieve sparse solutions. Another alternative step-down algorithm removes the variable if the corresponding $\hat{\theta}_j$ reaches zero at any step. The fourth algorithm is the one-step update algorithm, which uses the ESTIMATION and SELECTION stage only once, and do not pursue for the convergence. The performance of these four algorithms is compared in a simulation study in Section 5.2.2. Please refer to Section 2.3.1 for details on the algorithms.

3.4.2 Complete Algorithm for Adaptive Correlated COSSO

Compared to the Correlated COSSO, the complete Adaptive Correlated COSSO algorithm requires one more step to estimate the weights using an initial estimates of the functional components. The complete algorithm is following:

1. Fit an the initial model (a correlated SS-ANOVA model is recommended) and calculate the weights (w_j's) from the L_2 norms of each component.

2. Fix $\hat{\theta}_j = 1, \quad j = 1, \dots, q$.

3. Solve (3.7) for $\mathbf{c}, d, \tau, \sigma$ and λ_0^* using the ESTIMATION stage. Fix λ_0^* for the rest of the algorithm.

4. For each value in a reasonable grid of M, fix this parameter.

5. Use the algorithms above (full iteration, full iteration with truncation, step-down, one-step update) to fit Adaptive Correlated COSSO model.

6. Calculate the score of the selected tuning method.

7. Repeat steps 5 and 6 within the whole pre-selected tuning grid of M.

8. Find the minimal tuning score. The corresponding solution is the Adaptive Correlated COSSO estimate.

CHAPTER 4

COMPUTATIONAL ISSUES ON LARGE DATASETS

4.1 Challenges in Computation for Large Datasets

In general, nonparametric regression methods are computationally more intensive compared to their linear counterparts. Especially when the dimension of the dataset gets larger, the computation time is often dramatically increased due to the curse of dimensionality. We have already mentioned about the computational difficulties for the correlated SS-ANOVA models when the number of explanatory variables increases, mainly due to multi-dimensional smoothing parameter selection. As shown in previous two chapters, both methods we proposed in this dissertation (Correlated COSSO and Adaptive Correlated COSSO) overcome this issue since they include only one smoothing parameter in their formulations.

In practice, we use a grid search to select the single smoothing parameter. For each value in the grid, the proposed computational algorithms of Section 2.3.1 (or Section 3.4.1 for Adaptive Cor-COSSO) namely full iteration, full iteration with truncation, Step-Down and One-Step Update algorithms can be implemented. Any of these four algorithms use ESTIMATION and SELECTION stages either only once, or multiple times. The SELECTION stage solves a *q*-

dimensional quadratic programming problem, where q is the number of components in the full model. On the other hand, the ESTIMATION stage needs solution of the SS-ANOVA which takes more time when the number of observations increases. Therefore, the computation time for the SELECTION stage is negligible compared to the time spent by the ESTIMATION stage, and the ESTIMATION stage is contributing the main part to the computational time. The main reason for this is that we need to fit a linear mixed effects model at the ESTIMATION stage, which requires the estimation of n random effects, where n is the number of observations in the dataset. This actually is a big improvement compared to the correlated SS-ANOVA estimation of Wang (1998b), since the linear mixed model in their formulation contains $n \times q$ random effects. Both Correlated COSSO and Adaptive Correlated COSSO decrease the computation time considerably compared to SS-ANOVA when the number of components is large. However, when the number of observations is large, the computation will still be expensive.

In this chapter, we propose an alternative algorithm: Subset Basis Algorithm to further decrease the computational time in these two methods especially when the sample size is large. The main idea of this algorithm is to reduce the dimensionality of the basis functions to be used for function estimation. The number of random effects to be estimated in the ESTIMATION stage will be automatically decreased with this new algorithm.

Parsimonious basis approach has been used in nonparametric regression literature (Xiang and Wahba, 2006; Ruppert and Carroll, 2000; Yau, Kohn, and Wood, 2002). These methods use a subset of the observations at hand. The idea of the subset basis algorithm is to minimize the objective function in Correlated COSSO problem (2.5) in a subspace of $\mathscr{H}$ spanned by a smaller number of basis functions. In the full basis algorithm, all n observations are used to create one set of basis functions. However, in subset basis algorithm, N basis functions (usually $N < n$) are used. When N is smaller, the computation time decreases more; on the other hand, the approximation is expected to be better when we use a larger number of basis functions. We would like N to be as small as possible, yet still provide a good approximation. We try to find a practical answer to this question with a simulation study.

One question raised in the literature about the subset basis algorithms is how to select the subset from observations to form a set of basis. Simple random sampling of data points is in common use. Alternatively, Xiang and Wahba (2006) recommended using a cluster algorithm to sample the subset. Details on the sampling methods are given in Section 4.2.3. We use both simple random and cluster sampling methods, and compare their performances in a simulation study. The results from this simulation study can be found in Section 5.2.3.

Zhang and Lin (2006) applied the subset basis algorithm to original COSSO method, which has successfully decreased the computation time for COSSO in exponential families. Extensive simulation examples in their paper suggest that the subset basis algorithm performs almost as good as the full basis algorithm. We would like to follow a similar approach for Correlated COSSO.

The rest of the chapter is organized as follows. In Section 4.2.1, we provide the formulation of the subset basis algorithm. We use the Correlated COSSO method when presenting the algorithm, yet, the algorithm can also be applied to the Adaptive Correlated COSSO method with some minor changes. The algorithm for a fixed M is illustrated in Section 4.2.2. The differences of subset basis algorithm from the full basis algorithm will be clearly stated in both SELECTION and ESTIMATION stages. To be more specific, the linear mixed model used will be completely altered, which will lead to a different model from that in the full basis algorithm. Section 4.2.3 covers the details of two sampling methods: simple random sampling and cluster sampling. All methods require the influence matrix, and the computation of this matrix is different for the subset basis algorithm. We address this issue in Section 4.2.4. In Section 4.3 we provide the complete subset basis algorithm for solving the Correlated COSSO problem.

4.2 Subset Basis Algorithm

4.2.1 Subset Basis Formulation

In this section, we present the formulation of the subset basis algorithm for both Correlated COSSO and Adaptive Correlated COSSO. The algorithm is based on choosing a subset of observations and creating a set of basis functions using this subset.

With a given sampling scheme, we randomly select N points from n observations of the dataset, denoted as $\{\mathbf{x}_{1*}, \ldots, \mathbf{x}_{N*}\}$, and use these observations to generate N basis functions. We will find the minimizer of Cor-COSSO (or Adaptive Cor-COSSO) solution using the basis functions constructed from the subset. In other words, we assume the minimizer of (2.5) of

Cor-COSSO with the following form:

$$f(\mathbf{x}) = d + \sum_{i=1}^{N} c_i R_\theta(\mathbf{x}_{i*}, \mathbf{x}) = d + \sum_{i=1}^{N} c_i \sum_{j=1}^{q} \theta_j R_j(\mathbf{x}_{i*}, \mathbf{x}), \tag{4.1}$$

where $R_j(\cdot,\cdot)$ is defined in earlier chapters as the reproducing kernel of $\mathscr{H}^j$. We should point out that, since there are N basis functions in this new solution, $\hat{\mathbf{c}}$ is a N-dimensional vector instead of n-vector as in the full basis algorithm. Remark that $N \leq n$, and the subset basis algorithm will be useful if the subset size is smaller than the sample size.

Let $\mathbf{c} = (c_1, \ldots, c_N)^{\mathrm{T}}$, and $\mathbf{1}_n$ is an n-dimensional vector with each entry being 1. Let R_j^* be an $n \times N$ matrix with the entries $\{R_j(\mathbf{x}_i, \mathbf{x}_{k*})\}, i = 1, \ldots, n, k = 1, \ldots, N$. With some abuse of notation, we also define $R_\theta^* = \sum_{j=1}^{q} \theta_j R_j^*$. Remark that the dimension of R_θ^* is also $n \times N$. Furthermore, let R_j^{**} be an $N \times N$ matrix with the entries $\{R_j(\mathbf{x}_{i*}, \mathbf{x}_{k*})\}, i = 1, \ldots, N, k = 1, \ldots, N$, and define $R_\theta^{**} = \sum_{j=1}^{q} \theta_j R_j^{**}$.

The solution (4.1) can then be written in matrix format:

$$\mathbf{f}_j = \theta_j R_j^* \mathbf{c}, \quad j = 1, \ldots, q, \quad \mathbf{f} = \mathbf{1}d + \sum_{j=1}^{q} \mathbf{f}_j = \mathbf{1}d + R_\theta^* \mathbf{c},$$

and the penalty term in the equivalent form becomes:

$$\sum_{j=1}^{q} \theta_j^{-1} \|P^j f\|^2 = \sum_{j=1}^{q} \theta_j \mathbf{c}^{\mathrm{T}} R_j^{**} \mathbf{c} = \mathbf{c}^{\mathrm{T}} R_\theta^{**} \mathbf{c}.$$

The Correlated COSSO optimization problem (2.6) becomes:

$$\min_{d,\mathbf{c},\theta\geq\mathbf{0}} \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{1}_n d - R^*_\theta\mathbf{c})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{1}_n d - R^*_\theta\mathbf{c}) + n\lambda_0\mathbf{c}^{\mathrm{T}}R^{**}_\theta\mathbf{c} + n\lambda_1\mathbf{1}_q^{\mathrm{T}}\theta. \tag{4.2}$$

4.2.2 Computation with Subset Basis Algorithm

The minimizer of (4.2) can be calculated using an iterative approach. At this moment, we assume λ_1 to be fixed. Tuning of λ_1 will be necessary, and this will be covered in Section 4.2.4.

The computational algorithm will use the following two stages as the building blocks:

1. **ESTIMATION** Stage: With θ fixed, solve $(d,\mathbf{c},\tau,\sigma^2)$ using the following procedure.

   With $(\sigma^2,\tau,\lambda_0)$ fixed, we solve penalized weighted least squares problem:

   $$\min_{d,\mathbf{c},} \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{1}_n d - R^*_\theta\mathbf{c})^{\mathrm{T}}\mathbf{W}_\tau(\mathbf{y}-\mathbf{1}_n d - R^*_\theta\mathbf{c}) + n\lambda_0\mathbf{c}^{\mathrm{T}}R^{**}_\theta\mathbf{c}.$$

   We treat $(\sigma^2,\tau,\lambda_0)$ as variance components, and estimate them using Generalized Maximum Likelihood method. Simultaneous estimation of these parameters is done with linear mixed model connection.

   Define the solution as $(\hat{d},\hat{\mathbf{c}},\hat{\tau},\hat{\sigma^2})$.

2. **SELECTION** Stage: Let $\lambda_0^* = 2\lambda_0\sigma^2$. With $(d,\mathbf{c},\tau,\sigma^2)$ fixed at $(\hat{d},\hat{\mathbf{c}},\hat{\tau},\hat{\sigma^2})$, the solu-

tions from ESTIMATION stage, solve the optimization problem for θ:

$$\min_{\theta} \left(\mathbf{y} - \mathbf{1}_n \hat{d} - R^*_{\theta}\hat{\mathbf{c}}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \left(\mathbf{y} - \mathbf{1}_n \hat{d} - R^*_{\theta}\hat{\mathbf{c}}\right) + n\lambda_0^* \hat{\mathbf{c}}^{\mathrm{T}} R^{**}_{\theta} \hat{\mathbf{c}}$$

subject to $\sum_{j=1}^{q} \theta_j \leq M, \quad \theta_j \geq 0, \quad j = 1, \ldots, q.$

Here $M \geq 0$ is the tuning parameter, which is one-to-one corresponding to λ_1. Details on two stages are following.

ESTIMATION Stage:

In the ESTIMATION stage, we fix θ, and then the last quantity in (4.2) disappears. Also taking τ, σ^2 and λ_0 as fixed, the problem becomes finding a minimizer for

$$\min_{d,\mathbf{c}} (\mathbf{y} - \mathbf{1}_n d - R^*_{\theta}\mathbf{c})^{\mathrm{T}} \mathbf{W}_{\tau} (\mathbf{y} - \mathbf{1}_n d - R^*_{\theta}\mathbf{c}) + n\lambda_0^* \mathbf{c}^{\mathrm{T}} R^{**}_{\theta} \mathbf{c}. \tag{4.3}$$

where $\lambda_0^* = 2\sigma^2\lambda_0$. This is the same as the SS-ANOVA formulation with subset basis. We would like to find a solution in the form $\hat{\mathbf{f}} = \mathbf{1}_n \hat{d} + R^*_{\theta}\hat{\mathbf{c}}$. Taking the derivative of (4.3) with respect to d and $\mathbf{c}$ and equating them to zero will result in the following equation system:

$$\begin{aligned} \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_{\tau} R^*_{\theta} \mathbf{c} + \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_{\tau} \mathbf{1}_n d &= \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_{\tau} \mathbf{y}, \\ \left(R^{*\mathrm{T}}_{\theta} \mathbf{W}_{\tau} R^*_{\theta} + n\lambda_0^* R^{**}_{\theta}\right) \mathbf{c} + R^{*\mathrm{T}}_{\theta} \mathbf{W}_{\tau} \mathbf{1}_d &= R^{*\mathrm{T}}_{\theta} \mathbf{W}_{\tau} \mathbf{y}. \end{aligned} \tag{4.4}$$

The main difference between this stage and the ESTIMATION stage of the full basis algorithms (see Section 2.3.1) is that, the reproducing kernel matrix R_θ in Equation (2.7) is replaced by two different matrices: by R^*_θ in the first part of the equation controlling the bias of the fit, and by R^{**}_θ in the penalty term. This brings extra difficulty in implementing the Correlated COSSO (and Adaptive Cor-COSSO) method. The linear mixed effects model (2.9) cannot be modified to provide the matrix representation in (4.4). Instead, consider the following linear mixed model:

$$\mathbf{y} = \mathbf{1}_n d + R^*_\theta \mathbf{u} + \varepsilon, \tag{4.5}$$

where $\mathbf{u} \sim N\left(\mathbf{0}, \frac{\sigma^2}{n\lambda_0^*}(R^{**}_\theta)^+\right)$ and $\varepsilon \sim N\left(\mathbf{0}, \sigma^2 \mathbf{W}_\tau^{-1}\right)$. Here $(R^{**}_\theta)^+$ is the Moore Penrose inverse of R^{**}_θ. Reader should remark that $\mathbf{u}$ is an N dimensional vector, which means the number of random effects contained in this model is decreased from n to N. The estimate of $\mathbf{u}$ is $\hat{\mathbf{u}} = \hat{\mathbf{c}}$, and the smoothing spline estimate is $\hat{\mathbf{f}} = \mathbf{1}_n \hat{d} + R^*_\theta \hat{\mathbf{c}} = \mathbf{1}_n \hat{d} + R^*_\theta \hat{\mathbf{u}}$. Equation (3.3) of Harville (1977) provides the following normal equations for the mixed model:

$$\begin{pmatrix} \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_\tau \mathbf{1}_n & \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_\tau R^*_\theta \\ R^*_\theta \mathbf{W}_\tau \mathbf{1}_n & R^{*\mathrm{T}}_\theta \mathbf{W}_\tau R^*_\theta + n\lambda_0^* R^{**}_\theta \end{pmatrix} \begin{pmatrix} d \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{1}_n^{\mathrm{T}} \mathbf{W}_\tau \mathbf{y} \\ R^{*\mathrm{T}}_\theta \mathbf{W}_\tau \mathbf{y} \end{pmatrix}. \tag{4.6}$$

The equation system above is identical to that in (4.4). Smoothing parameter λ_0^*, and variance covariance parameters σ^2 and τ are REML estimates as already discussed in Chapter 2.

As stated above, there are N random effects to be estimated in the mixed model in ESTIMATION stage. We would like to remind the reader that the mixed effects model fit is the most time consuming part in the Correlated COSSO (and the Adaptive Cor-COSSO) algorithm. Decreasing the number of random effects will dramatically reduce the computation time at the ESTIMATION stage, hence the whole algorithm. In a massive dataset with lots of observations, the estimation of Correlated COSSO with the full basis algorithm might be very slow. Yet, with the subset basis algorithm, we can take a pre-specified smaller number of observations to create the basis subset, which will decrease the computation time considerably.

One difficulty we faced in the practical implementation of ESTIMATION stage with the subset basis algorithm is the Moore Penrose inverse of R_θ^{**} matrix. In theory, the reproducing kernel is a positive definite matrix, and the inversion of this matrix should not be a problem. However, from time to time we observe the R_θ^{**} matrix to be very close to singularity, which results in very small eigenvalues. The presence of these tiny eigenvalues can cause the inversion to be numerically unstable. A practical solution we propose in this case is to add a small positive diagonal matrix to R_θ^{**} to make it better conditioned. In other words, we essentially calculate $R_\theta^{**} + \alpha I$, where α is a small positive constant (for example, $\alpha = 0.0001$) is used in our simulation study.

We point out that the Adaptive Correlated COSSO solution can be achieved using basis subset algorithm as well. In particular, we need to re-arrange the R_θ^* and R_θ^{**} matrices in order to calculate the adaptive kernel matrices. To be specific, define $K_{\mathbf{w},\theta}^* = \sum_{j=1}^q \theta_j w_j^{-2} R_j^*$ be an $n \times N$ matrix, and $K_{\mathbf{w},\theta}^{**} = \sum_{j=1}^q \theta_j w_j^{-2} R_j^{**}$ be an $N \times N$ matrix. The only thing to change for Adaptive Cor-COSSO method is to replace R_θ^* by $K_{\mathbf{w},\theta}^*$ and R_θ^{**} by $K_{\mathbf{w},\theta}^{**}$ in the formulation above.

SELECTION Stage:

In this stage, the function estimate and the variance-covariance parameters are fixed at their current values, and the minimization takes place over θ parameters. When $(\hat{d}, \hat{\mathbf{c}}, \hat{\tau}, \hat{\sigma^2}, \lambda_0^*)$ are fixed, then the equivalent formulation to Correlated COSSO becomes:

$$\min_{\theta} \left(\mathbf{y} - \mathbf{1}_n\hat{d} - R_\theta^*\hat{\mathbf{c}}\right)^{\mathrm{T}} \mathbf{W}_{\hat{\tau}} \left(\mathbf{y} - \mathbf{1}_n\hat{d} - R_\theta^*\hat{\mathbf{c}}\right) + n\lambda_0^*\hat{\mathbf{c}}^{\mathrm{T}} R_\theta^{**}\hat{\mathbf{c}} \tag{4.7}$$

subject to $\sum_{j=1}^{q} \theta_j \leq M, \quad \theta_j \geq 0, \quad j = 1, \ldots, q.$

Denote $g_j^* = R_j^*\hat{\mathbf{c}}$ for $j = 1, \ldots, q$, and $\mathbf{G}^*$ as an $n \times q$ matrix with j^{th} column being g_j^*. Let $g_j^{**} = R_j^{**}\hat{\mathbf{c}}$ for $j = 1, \ldots, q$, and $\mathbf{G}^{**}$ as an $N \times q$ matrix with j^{th} column being g_j^{**}. Also let $\mathbf{H} = \mathbf{G}^{*\mathrm{T}}\mathbf{W}_{\hat{\tau}}\mathbf{G}^*$, and $\mathbf{a} = (n\lambda_0^*/2)\mathbf{G}^{**\mathrm{T}}\hat{\mathbf{c}} - \mathbf{G}^{*\mathrm{T}}\mathbf{W}_{\hat{\tau}}(\mathbf{y} - \mathbf{1}_n\hat{d})$. A similar argument to Cor-COSSO SELECTION stage shows that (4.7) is equivalent to:

$$\min_{\theta} \frac{1}{2}\theta^{\mathrm{T}}\mathbf{H}\theta + \mathbf{a}^{\mathrm{T}}\theta, \quad \text{subject to} \quad \mathbf{1}_q^{\mathrm{T}}\theta \leq M, \quad \theta_j \geq 0 \quad j = 1, \ldots, q \tag{4.8}$$

which is a Quadratic Programming (QP) problem.

For Adaptive Correlated COSSO, the matrices $\mathbf{G}^*$ and $\mathbf{G}^{**}$ need to be redefined. Let $g_j^* = w_j^{-2}K_{\mathbf{w},\theta}^*\hat{\mathbf{c}}$ for $j = 1, \ldots, q$, and $\mathbf{G}^*$ as an $n \times q$ matrix with j^{th} column being g_j^*, $g_j^{**} = w_j^{-2}K_{\mathbf{w},\theta}^{**}\hat{\mathbf{c}}$ for $j = 1, \ldots, q$, and $\mathbf{G}^{**}$ as an $N \times q$ matrix with j^{th} column being g_j^{**}. The remaining part for the SELECTION stage is the same as above.

4.2.3 Sampling Methods for Basis Selection

The main idea of subset basis algorithm is to use a subset of the design points to create a set of basis functions for model estimation. In other words, we would like this subset to be representative of the dataset. One important issue is to find a good sampling scheme to select these points automatically from the whole data. In this section, we consider two sampling methods, which will be used to choose N data points from the total n observations.

The first method is the simple random sampling (SRS hereafter). In SRS each individual has the same probability of being selected in the subset. The motivation behind is that we assume the data has a uniform distribution in the design space, therefore the random sampling of the observations will provide a good representation of the whole design space. We use SRS without replacement, hence any observation can be selected to the subset only once. The most appealing reason for using SRS in subset basis algorithm is its ease of implementation. It has already been applied in some parsimonious basis algorithms (see Ruppert and Carroll 2000, Yau, Kohn, and Wood 2002, Zhang and Lin 2006). The implementation of the method is straightforward.

Simple random sampling randomly selects data points without taking any information about their distance from each other. Therefore, there might be observations in the subset which are very close to each other. One alternative to take this distance into account is to group the design points into N separate classes and from each of these classes select one observation to the subset.

The method just mentioned can be implemented using the idea of clustering. In the subset basis algorithm with clustering, we recommend separating the data set into N clusters, and then randomly selecting one observation from each cluster. There are several available algorithms for clustering. We use k-means clustering method with the Euclidean distance as the distance measure. We use the SAS software as our computational tool for the Cor-COSSO and Adaptive Cor-COSSO. For cluster sampling, we choose a fast clustering algorithm (FASTCLUS, SAS/STAT User's Guide, SAS Institute Inc.) available in this software package. The advantage of the FASTCLUS algorithm is that it implements the clustering to large datasets in a very short time. The clustering-based subset basis algorithm works in the following four steps:

1. Select N points as cluster seeds.

2. Assign all design points to some cluster based on their nearest seed, and replace the cluster seed by the cluster mean. Repeat this step until change is smaller than a predefine convergence limit.

3. Form the final clusters based on the last iteration

4. Randomly select one design point from each cluster. This set of points will create the subset to fit the Cor-COSSO model.

4.2.4 Tuning in Subset Basis Algorithm

In the subset basis algorithm, we can use any of the four tuning criteria described in Section 2.3.2, namely: the WMSE, UBR GCV and the L methods. These criteria are formulated using the *influence* or *hat* matrix (**A**), which was calculated using the QR decomposition of $\mathbf{1}_n$ shown in Section 1.3.3. However, the *hat* matrix does not have the same form in the subset basis algorithm, and therefore, the calculation of the tuning scores will be different than that in full basis algorithm. In other words, we can still use the tuning criteria mentioned above, but we still need to figure out the matrix formulation of the **A**.

The subset basis algorithm minimization problem of (4.3) can be rephrased in the matrix format as:

$$\left(\mathbf{y}-\begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ d \end{pmatrix}\right)^{\mathrm{T}} \mathbf{W}_\tau \left(\mathbf{y}-\begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ d \end{pmatrix}\right) + n\lambda^*_0 \begin{pmatrix} \mathbf{c} & d \end{pmatrix}\begin{pmatrix} R^{**}_\theta & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ d \end{pmatrix} \tag{4.9}$$

Taking derivatives of (4.9) with respect to **c** and d:

$$\begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix}^{\mathrm{T}} \mathbf{W}_\tau \left(\mathbf{y}-\begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ d \end{pmatrix}\right) - n\lambda^*_0 \begin{pmatrix} R^{**}_\theta & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ d \end{pmatrix} = \mathbf{0}_{N+1}$$

Now, let $\mathbf{T} = \begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix}^{\mathrm{T}} \mathbf{W}_\tau \begin{pmatrix} R^*_\theta & \mathbf{1}_n \end{pmatrix} + n\lambda^*_0 \begin{pmatrix} R^{**}_\theta & 0 \\ 0 & 0 \end{pmatrix}$. Then the estimates of $\hat{\mathbf{c}}$ and

$\hat{d}$ can be written as:

$$\begin{pmatrix} \hat{\mathbf{c}} \\ \hat{d} \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} R_\theta^* & \mathbf{1}_n \end{pmatrix}^{\mathrm{T}} \mathbf{W}_\tau \mathbf{y}$$

Hence, the *hat* matrix **A** is;

$$\mathbf{A} = \begin{pmatrix} R_\theta^* & \mathbf{1}_n \end{pmatrix} \mathbf{T}^{-1} \begin{pmatrix} R_\theta^* & \mathbf{1}_n \end{pmatrix}^{\mathrm{T}} \mathbf{W}_\tau \tag{4.10}$$

In the following section, we present the complete algorithm with subset basis approach for the Correlated COSSO method. The algorithm will be similar for the Adaptive Correlated COSSO.

4.3 Complete Algorithm for Subset Basis Approach

In this section we present the basis subset algorithm for the Correlated COSSO. We use the One-Step Update algorithm for each value of M on a pre-defined tuning grid. User has the option to use the other three alternative algorithms (i.e., full iteration, full iteration with truncation and Step-Down). Moreover, for tuning M, we recommend the use of L method. The alternative tuning methods such as GCV, UBR and WMSE can be also used.

The complete subset basis algorithm for the Correlated COSSO is following:

1. Select a subset of N observations randomly from the data. Create the subset basis by the reproducing kernels $R_j(\cdot,\cdot)$ for $j = 1,\ldots,q$.

2. Fix $\hat{\theta}_j = 1, \quad j = 1, \ldots, q$.
3. Solve (4.3) for $\mathbf{c}, d, \tau, \sigma$ and λ_0^* using the ESTIMATION stage. Fix λ_0^* for the rest of the algorithm.
4. For each value in a reasonable grid of M, fix this parameter.
5. Use one-step update algorithm of Section 2.3 to fit Cor-COSSO model.
6. Calculate the L method score.
7. Repeat steps 5 and 6 within the grid of M.
8. Find the minimal L score. The corresponding solution is the Correlated COSSO estimate.

The algorithm can easily be adapted to the Adaptive Correlated COSSO method. The implementation of this method will be provided with a simulation study in Section 5.2.3.

CHAPTER 5

SIMULATION STUDY

5.1 Introduction

In this chapter, we outline the simulation designs and summarize our findings. This chapter consists of two parts. The first part is designed to compare various tuning methods for both Cor-COSSO and Adaptive Cor-COSSO. We also compare the proposed algorithms for fixed tuning parameter, sampling schemes and the subset basis versus full basis approaches. The second part compares Correlated COSSO and Adaptive Correlated COSSO with existing methods in the literature.

The empirical performance of the Correlated COSSO and the Adaptive Correlated COSSO methods are studied and demonstrated in this section using extensive simulations. We present simulations in the following settings:

- error covariance structure: AR(1) and Compound Symmetry (CS),
- sample size: $n = 100, 200, 400, 1000$,
- dimension of covariates: $p = 10, 28, 30$,

- different error standard deviations: $\sigma = 0.5, 1, 2, 4$,

- different signal to noise ratio (SNR): $0.9 - 7.2$.

Our numerical examples also cover scenarios where the explanatory variables are independent or correlated, highly correlated data situations, a large dimensional variable selection example, and a model including interaction terms.

We use the original COSSO as the benchmark method to compare with our proposals. Lin and Zhang (2006) and Zhang and Lin (2006) shows that the original COSSO method outperforms MARS (Friedman, 1991) in their simulation study, which is one of the most popular component selection methods in nonparametric regression. Although the original COSSO does not take the error covariance into account, it is still a good method to compare with for our proposals.

The following four functions are used as building blocks in our simulation study:

$$g_1(t) = t, \quad g_2(t) = (2t-1)^2, \quad g_3(t) = \frac{sin(2\pi t)}{2 - sin(2\pi t)},$$

$$g_4(t) = 0.1sin(2\pi t) + 0.2cos(2\pi t) + 0.3sin(2\pi t)^2 + 0.4cos(2\pi t)^2 + 0.5sin(2\pi t)^3.$$

These four functions are used to generate the relationship in between the explanatory variables and the response.

Throughout the simulation study, in order to calculate the Signal-to-Noise Ratio (SNR) defined as:

$$SNR = \sqrt{\frac{Var[f(x)]}{|V|^{1/n}}}$$

We generate Monte Carlo samples to estimate the variances of f and noise term separately. In particular, we generate 10,000 observations to estimate $Var[f(x)]$, and the covariance matrix $V = Cov(\varepsilon)$. There are two possible approaches to estimate the magnitude of noise: the first one is based on the trace, and the second is based on the determinant of the covariance matrix V. Since the trace-based method ignores the correlation between observations, we choose to use the determinant-based method.

We measure the empirical performance of proposed methods in terms of their estimation accuracy and model selection. The Integrated Squared Error (ISE) is used as the measure of prediction accuracy. In order to calculate this quantity, a test set of size 10,000 is generated from the same regression function from which the data was generated. Using Monte Carlo integration, the ISE is estimated for each replication. Average and standard error estimates of ISE are reported.

We use three quantities to measure the model selection performance. The first quantity is the *correct model selection percentage* (π_c), which is the percentage of the correct model being identified by the method over 100 simulations. The other two quantities are the *Number of correct 0's*, (CORR) which is the number of uninformative variables which are successfully removed from the model, and the *Number of incorrect 0's*, (INC) which is the number of important variables left out of the model by mistake. Ideally, the last quantity should be close to 0. The averages and standard errors of these quantities from 100 simulated datasets are tabulated for each method. Abstract outlines of each simulation study is following.

Section 5.2.1 compares the smoothing parameter selection methods covered in Section 2.3.2. The methods include Weighted Mean Squared Error (WMSE), Generalized Cross Validation (GCV), Unbiased Risk (UBR) and *L* method. The details of these methods can be found in Section 2.3.2.

We proposed four different algorithms to find a solution to Cor-COSSO (or Adaptive Cor-COSSO) in Section 2.3.1. These algorithms are the full iteration, full iteration with truncation, step-down and one-step update algorithms. Section 5.2.2 compares these algorithms to recommend one of them in future fitting of Cor-COSSO and Adaptive Cor-COSSO.

We investigate the subset basis algorithm in Section 5.2.3. Two sampling methods, namely simple random sampling and cluster sampling, are compared and the selection of the basis size is also discussed based on the simulation results. The subset basis algorithms are compared to the full basis algorithm. We compare Cor-COSSO and Adaptive Cor-COSSO methods with the original COSSO in Section 5.3 using three simulation examples.

5.2 Comparison of Algorithms and Tuning Criteria

This section compares several tuning criteria and algorithms previously proposed in Chapters 2 - 4. We consider three examples. The first example covers the comparison of the tuning criteria described in Section 2.3.2. The second example is designed to compare four algorithms (full iteration, full iteration with truncation, step-down, one-step update). The third section is on the Subset Basis Algorithm (hereafter SBA) described in Chapter 4.

5.2.1 Comparison of Smoothing Parameter Selection Methods

We now compare the performance of various types of smoothing parameter selection scores described in Section 2.3.2. The tuning criteria under comparison are UBR's, GCV's, WMSE's and L method.

We consider a ten-dimensional additive model. The underlying data generating process is $f(x) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(3)}) + 6g_4(x^{(4)})$, where $g_1, \ldots, g_4$ are defined in the previous section. As can be seen from the process, $x^{(5)}, \ldots, x^{(10)}$ do not carry any information on the response, therefore they are *noise* variables. Let the sample size (n) be 200 and we generate the data by $y = f(x) + \varepsilon$, where ε's have mean 0 and constant standard deviation: $\sigma = 1$ and $\sigma = 2$. Furthermore, these error terms have a first-order Auto Regressive (AR-1) correlation structure, with the correlation parameter $\rho = 0.3$ (see $\mathbf{W}_{\tau}^{-1}$ matrix in Equation 1.18). A time series data is a motivation for this kind of a correlation structure. The signal-to-noise ratio (SNR) for this example is 1.86 when $\sigma = 2$ and 3.74 when $\sigma = 1$.

We fit the Additive Correlated COSSO to 100 simulated data sets and the results are reported. For one simulated dataset with $\sigma^2 = 1$, the magnitudes of the estimated components are plotted with the tuning parameter M (see Figure 5.1). These magnitudes are measured by L_1 norms, defined as $\frac{1}{n}\sum_{i=1}^{n} \|\hat{f}_j(x_i^{(j)})\| \quad$ for $j = 1, 2, \ldots, q$. For each tuning criteria, λ_0^* is es-

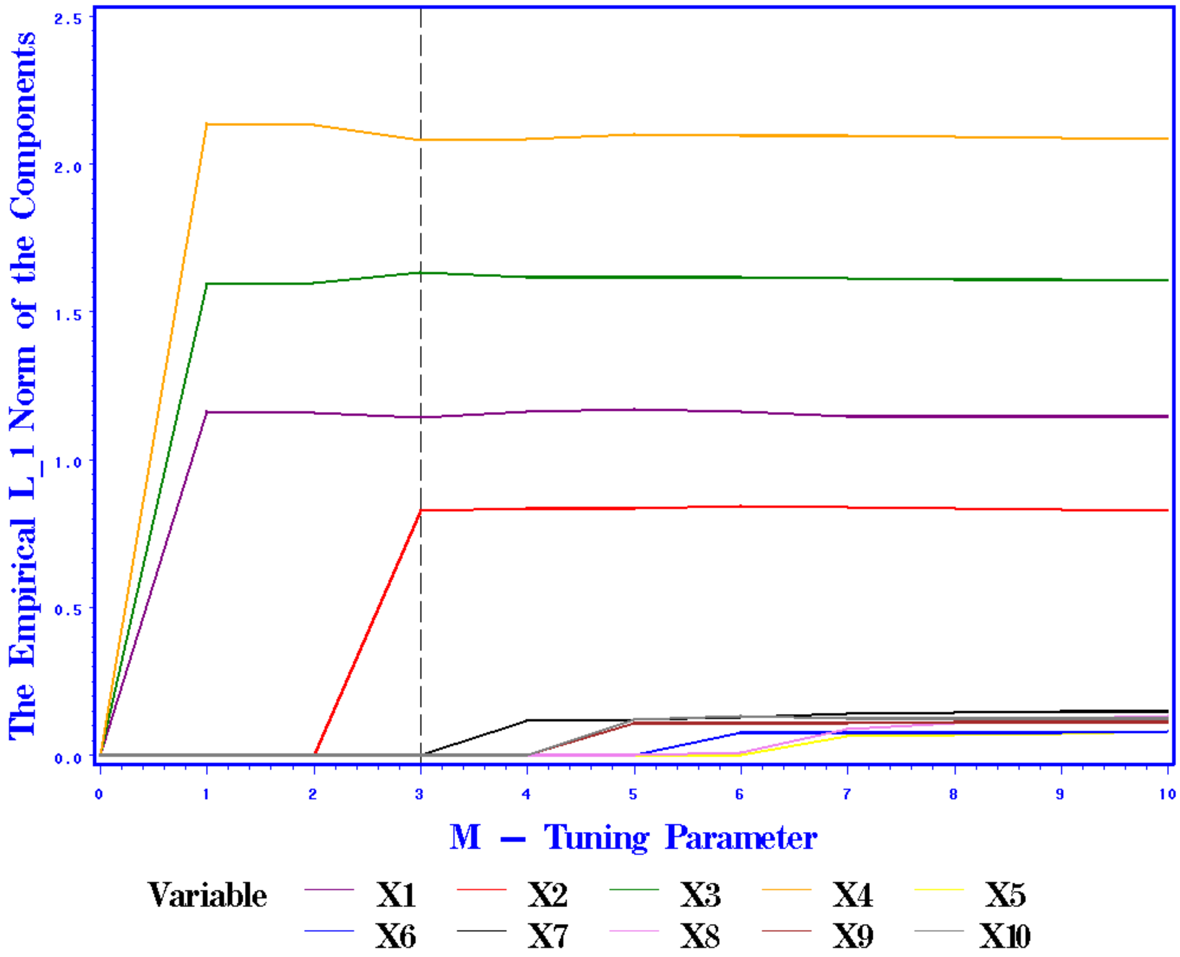

Figure 5.1: The empirical L_1 norm of the estimated components as plotted against the tuning parameter M.

timated in *ESTIMATION* stage using GML. The tuning of M is conducted on integers between 1 to 10. For this example we observe that the UBR0, UBR1, GCV0, L, and WMSE0 selected $M = 3$, which includes all four informative variables, and no noise variables. Furthermore, the GCV1, GCV2 and WMSE1 methods selected $M = 4$, and the UBR2 and WMSE2 selected $M = 5$, both including extra noise variables in the final model.

Tables 5.1 and 5.2 summarize the performance of different tuning methods in two scenarios: $\sigma = 1$ and $\sigma = 2$. Tables include variance estimates, ISEs, variable selection performance measures and their standard errors. The row TRUE corresponds to the SS-ANOVA fit when the true model is assumed to be known. The solution is obtained using the linear mixed model connection recommended by Wang (1998b). This fit serves as a gold standard for comparison among various methods. However, it is unavailable in real examples, since the true model is generally unknown.

Table 5.1: Tuning Criteria Comparison ($\sigma = 1$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
TRUE	.33(.08)	0.99(.06)	.13(.03)	100	6.00(.00)	.00(.00)
L	.33(.08)	1.00(.05)	.14(.04)	99	5.96(.40)	.00(.00)
UBR0	.36(.16)	1.12(.54)	.14(.04)	100	6.00(.00)	.00(.00)
UBR1	.33(.08)	1.00(.06)	.15(.09)	99	6.00(.00)	.01(.10)
UBR2	.19(.11)	1.21(.23)	.71(.64)	43	6.00(.00)	.61(.57)
GCV0	.33(.08)	0.99(.06)	.17(.06)	70	5.46(1.03)	.00(.00)
GCV1	.33(.08)	0.99(.06)	.16(.05)	78	5.67(.74)	.00(.00)
GCV2	.34(.08)	0.99(.06)	.16(.06)	74	5.56(.90)	.00(.00)
WMSE0	.34(.13)	1.08(.45)	.14(.04)	98	5.98(.14)	.00(.00)
WMSE1	.29(.08)	1.02(.09)	.24(.17)	55	3.95(2.69)	.03(.17)
WMSE2	.18(.08)	1.20(.23)	.75(.59)	14	4.32(2.61)	.57(.57)

Notes: This table corresponds to the results from the simulation with $n = 200$, $\rho = 0.3$ and $\sigma = 1$. Corresponding SNR is 3.74. The columns are estimates provided with standard errors (in parenthesis) from 100 Monte Carlo samples. Integrated Squared Error (ISE), correct model selection percentages (π_c), average number of correct zeros (CORR) and incorrect zeros (INC) are explained in introduction section for this chapter.

In Table 5.1, the UBR0, UBR1, WMSE0 and L method perform equally well. Their ISE values are close to the TRUE fit, and the fit given by the L method is overall the closest to the true fit. Also these four methods have correct model selection percentages over 95%, and most of them do not miss any important variables.

However, as Table 5.2 shows, when the variable selection problem becomes more challenging (with smaller SNR), the performances of both UBR and GCV methods get worse, while the L method, WMSE0 and WMSE1 are more robust to smaller SNR. Since the WMSE methods are not applicable in practice when we do not know the underlying data generating process, we recommend the L method as the default tuning method for M, and use it in the later examples.

Table 5.2: Tuning Criteria Comparison ($\sigma = 2$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
TRUE	.32(.08)	1.99(.11)	.42(.12)	100	6.00(.00)	.00(.00)
L	.32(.07)	2.02(.11)	.47(.18)	95	5.98(.14)	.03(.17)
UBR0	.30(.09)	2.09(.17)	.86(.74)	61	6.00(.00)	.43(.59)
UBR1	.24(.08)	2.28(.29)	1.79(1.21)	20	6.00(.00)	1.13(.72)
UBR2	.23(.07)	2.37(.28)	2.15(1.16)	07	6.00(.00)	1.35(.61)
GCV0	.32(.08)	2.00(.12)	.53(.19)	68	5.44(.98)	.00(.00)
GCV1	.32(.08)	2.00(.12)	.53(.19)	70	5.47(.97)	.00(.00)
GCV2	.33(.08)	2.00(.11)	.54(.19)	69	5.42(1.05)	.00(.00)
WMSE0	.32(.07)	2.02(.11)	.46(.12)	97	5.92(.61)	.00(.00)
WMSE1	.31(.07)	2.02(.11)	.47(.13)	90	5.74(.94)	.00(.00)
WMSE2	.27(.08)	2.08(.15)	.82(.52)	45	4.75(2.13)	.26(.48)

Notes: This table corresponds to the results from the simulation with $n = 200$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.86.

Similar conclusions can be derived in other simulation examples, which are not presented here. UBR0 method is one of the best methods when the SNR value is high. However, its performance is poor in higher variance situations. The L method performs consistently well in terms of both variable selection and model prediction.

5.2.2 Comparison of Algorithms

Both Correlated COSSO and Adaptive Correlated COSSO methods have one smoothing parameter (λ) in their formulations. In Sections 2.3.1 and 3.4.1, we propose four computational algorithms for both Cor-COSSO and Adaptive Cor-COSSO with fixed tuning parameter. Now we compare these algorithms for fitting Cor-COSSO (and Adaptive Cor-COSSO) in terms of their computation times and solution properties.

We consider a similar model with the previous example. The regression function to generate the relationship is $f(x) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(3)}) + 6g_4(x^{(4)})$, where $g_1, \ldots, g_4$ are defined in introduction section of this chapter. For diversity, we generate the error terms with compound symmetry correlation structure. Each subject has five equally correlated observations, and the data set includes observations from 20 subjects ($n = 100$). The within-subject correlation ρ is 0.3. The error standard deviation is taken as $\sigma = 2$, which corresponds to a SNR value of 1.76.

The four algorithms mentioned in Chapter 2 are respectively full iteration, truncated full iteration, step-down and one-step update algorithms. The first three iterate between the ESTIMATION and SELECTION stages until convergence, therefore, they are computationally more expensive. Since the computational time of the one-step update algorithm is shorter than the other three, and if it also gives compatible performance based on variable selection and estimation accuracy, we will recommend using the one-step update algorithm.

Table 5.3 provides the results for four algorithms based on 100 simulated datasets. We did

Table 5.3: Computation Time and Solution Property of Cor-COSSO Algorithms.

Algorithm	Duration (MM:SS)	ISE(se)	π_c	CORR(se)	INC(se)
One Step Upd	00:32	1.20(.47)	69	5.87(.54)	.19(.39)
Full	01:59	1.40(.46)	0	0.39(.83)	.01(.10)
Full - Trun	02:00	1.19(.45)	70	5.82(.46)	.17(.38)
Step-Down	01:59	1.18(.54)	70	5.83(.45)	.18(.39)

Notes: This table corresponds to the results from the Correlated COSSO algorithm comparison simulation with $n = 100$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.76. Here, duration is in minutes and seconds format, and is time for the algorithm to conclude.

not provide the variance component estimates since these estimates are all close to the true parameter values. It can be easily seen from the table that the computation time for the one-step update algorithm is much shorter than the other three methods. The reason for this is, on average, the convergence is achieved in four to five iterations, which means going through the ESTIMATION stage 30 to 40 times more for each fit. Furthermore, the estimation accuracy and the variable selection performances of the one-step update algorithm are almost as good as the other three. Therefore, we recommend using the one-step update algorithm as the default algorithm to solve the Correlated COSSO methods.

In Section 2.3.1 we have already mentioned about the numerical problem that causes the non-sparse solution for the full iteration algorithm, which is confirmed by our numerical results in Table 5.3. When we take a closer look at the final results, we discover that, the method gives tiny θ_j estimates which do not affect the ISE values too much, yet those noise variable are not excluded. Therefore, most of the time, the full iteration algorithm estimates a final model with 9 or all 10 variables included. We could not pinpoint the numerical issue which causes this problem, yet when we truncate small θ_j's to 0, (see truncated full iteration algorithm results), we see the gain in the variable selection performance.

Table 5.4: Computation Time and Solution Property of Adaptive Cor-COSSO Algorithms.

Algorithm	Duration (MM:SS)	ISE(se)	π_c	CORR(se)	INC(se)
One Step Upd	00:41	1.09(.45)	76	5.85(.41)	.12(.33)
Full	02:20	1.24(.46)	0	1.17(1.00)	.00(.00)
Full - Trun	02:21	1.10(.44)	69	5.73(.57)	.10(.30)
Step-Down	02:23	1.09(.43)	76	5.85(.39)	.10(.30)

Notes: This table corresponds to the results from the Adaptive Correlated COSSO algorithm comparison simulation with $n = 100$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.76. Here, duration is in minutes and seconds format, and is time for the algorithm to conclude.

In Table 5.4, results from the comparison of Adaptive Correlated COSSO algorithms can be found. Again the one-step update algorithm finds the solution in a shorter time with the results quite close to the truncated full and step-down algorithms. The sparsity issue of the full iteration algorithm is also noticed, and therefore the full iteration algorithm is not recommended for use in Adaptive Cor-COSSO.

5.2.3 Massive Data Example with Subset Basis Algorithm

The major drawback in the Correlated COSSO and Adaptive Correlated COSSO methods is their computational burden, when the number of observations (n) is large. This issue mainly arises from the fact that in ESTIMATION stage of the algorithms, where we need to estimate n random effects, which becomes computationally challenging when the sample size increases.

In Chapter 4 we propose to use the Subset Basis Algorithm (hereafter SBA) for large datasets. The method uses a subset of samples to create a basis set, and uses this set to approximate the Correlated COSSO (or Adaptive Correlated COSSO) solution. In this section, we

Table 5.5: Subset Basis Algorithm with $n = 200$ Observations ($\sigma = 2$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.27(.08)	2.03(.13)	.73(.28)	84	5.96	.20	0:00:07
SBA-CL	25	.26(.09)	2.08(.13)	.95(.36)	54	5.86	.33	0:00:07
SBA-SRS	50	.27(.08)	2.01(.12)	.63(.23)	90	5.92	.05	0:00:09
SBA-CL	50	.27(.09)	2.02(.12)	.66(.27)	86	5.94	.09	0:00:09
SBA-SRS	100	.27(.09)	2.04(.11)	.65(.27)	85	5.93	.09	0:00:18
SBA-CL	100	.26(.10)	2.04(.11)	.65(.27)	84	5.91	.09	0:00:19
Full	200	.29(.09)	1.98(.13)	.58(.26)	89	5.95	.08	0:00:57

Notes: This table corresponds to the results from the simulation with $n = 200$, $\rho = 0.3$ and $\sigma = 2$. SBA-SRS stands for the Subset Basis Algorithm with Simple Random Sampling, while SBA-CS is with Cluster Sampling. Full method provides the results from One-Step Update algorithm for the Cor-COSSO with mixed model formulation in (2.9). Corresponding SNR is 1.80.

investigate the empirical performance of SBA. Although the method is applicable to Adaptive Cor-COSSO as well, we present results from only Cor-COSSO method.

Table 5.6: Subset Basis Algorithm with $n = 200$ Observations ($\sigma = 4$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.27(.08)	4.06(.26)	2.62(1.06)	16	5.91	.89	0:00:07
SBA-CL	25	.28(.08)	4.06(.25)	2.62(1.07)	12	5.88	.94	0:00:06
SBA-SRS	50	.28(.08)	4.03(.26)	2.29(.96)	21	5.90	.79	0:00:09
SBA-CL	50	.28(.08)	4.05(.25)	2.42(.94)	20	5.94	.83	0:00:09
SBA-SRS	100	.28(.08)	4.03(.26)	2.42(1.18)	23	5.81	.79	0:00:19
SBA-CL	100	.28(.08)	4.03(.25)	2.44(1.18)	24	5.80	.78	0:00:18
Full	200	.28(.08)	4.03(.26)	2.41(1.23)	20	5.94	.88	0:01:06

Notes: This table corresponds to the results from the simulation with $n = 200$, $\rho = 0.3$ and $\sigma = 4$. Corresponding SNR is 0.90.

Recall that we introduced two sampling methods, namely Simple Random Sampling (SRS) and Cluster Sampling (CL). Both methods are used to fit the Cor-COSSO with SBA. We compare the performance of SBA algorithm with varying basis size in several simulation settings. In these settings, we use different sample sizes ($n = 200, 400$ and 1000), and error standard deviations ($\sigma = 2,4$). We generate data using the regression function:

$$f(x) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(3)}) + 6g_4(x^{(4)})$$

with additive error terms with a compound symmetry correlation structure. In this example we assume 5 observation per subject and all observations from the same subject is equally correlated pairwise. The within-subject correlation is $\rho = 0.3$. The datasets also have six noise variables ($x^{(5)}, \dots, x^{(10)}$). The variable selection and prediction accuracy performance of the SBA settings along with the average computation time can be found in Tables 5.5 - 5.10. We provide results from both sampling methods and basis sizes $N = 25, 50, 100, 200, n$, where n is the sample size.

Table 5.7: Subset Basis Algorithm with $n = 400$ Observations ($\sigma = 2$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.29(.06)	2.03(.08)	.35(.08)	100	6.00	.00	0:00:10
SBA-CL	25	.28(.06)	2.06(.08)	.49(.13)	97	5.95	.01	0:00:11
SBA-SRS	50	.28(.07)	2.02(.08)	.32(.08)	99	5.97	.00	0:00:13
SBA-CL	50	.30(.06)	2.01(.08)	.32(.08)	100	6.00	.00	0:00:13
SBA-SRS	100	.30(.06)	2.00(.08)	.28(.08)	100	6.00	.00	0:00:26
SBA-CL	100	.30(.06)	2.01(.08)	.30(.08)	98	5.97	.00	0:00:26
SBA-SRS	200	.29(.06)	2.00(.08)	.27(.08)	100	6.00	.00	0:01:09
SBA-CL	200	.29(.06)	2.00(.08)	.28(.08)	100	6.00	.00	0:01:09
Full	400	.30(.06)	1.98(.08)	.26(.07)	99	5.99	.00	0:05:34

Notes: This table corresponds to the results from the simulation with $n = 400$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.87.

It can be seen from Table 5.5, the results of SBA are comparable with the full basis algorithm, especially when the basis size is 50 or above. ISE values are within small differences

Table 5.8: Subset Basis Algorithm with $n = 400$ Observations ($\sigma = 4$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.29(.06)	4.05(.17)	1.38(.42)	52	5.86	.39	0:00:10
SBA-CL	25	.29(.06)	4.04(.17)	1.22(.35)	66	5.97	.32	0:00:10
SBA-SRS	50	.29(.06)	4.06(.17)	1.37(.41)	61	5.94	.33	0:00:13
SBA-CL	50	.29(.06)	4.02(.16)	1.14(.38)	64	5.96	.32	0:00:14
SBA-SRS	100	.30(.06)	4.02(.17)	1.15(.39)	62	5.85	.27	0:00:25
SBA-CL	100	.30(.06)	4.02(.17)	1.13(.36)	61	5.90	.31	0:00:25
SBA-SRS	200	.30(.06)	4.02(.17)	1.15(.48)	60	5.90	.32	0:01:08
SBA-CL	200	.30(.06)	4.02(.16)	1.10(.39)	66	5.91	.28	0:01:09
Full	400	.30(.06)	4.00(.17)	1.08(.43)	60	5.94	.37	0:06:02

Notes: This table corresponds to the results from the simulation with $n = 400$, $\rho = 0.3$ and $\sigma = 4$. Corresponding SNR is 0.94.

from the original model, and variable selection performance is similar. On the other hand, the SBA with basis size 25 shows instable results.

In Figure 5.2, the estimated functional components are plotted along with the true functional component, the curves are from subset basis algorithm fits with basis sizes 25, 50, 100, and the full basis algorithm. Notice the components are centered according to the ANOVA decompositions. Overall, all four components SBA fits and the full basis algorithm fit are quite close to the real functional component. The function estimates from SBA with $N = 25$ shows the largest divergence from the true fit as expected. The best fit is provided by the full basis algorithm, yet the SBA fits with $N = 50$ and $N = 100$ give good approximations to this fit.

In Table 5.6 the error standard deviation is increased to 4, hence the problem becomes more challenging. We can see that except for basis size 25, the results from SBA are close to the full basis algorithm. Error standard deviation ($\hat{\sigma}$) and correlation ($\hat{\rho}$) estimates are close

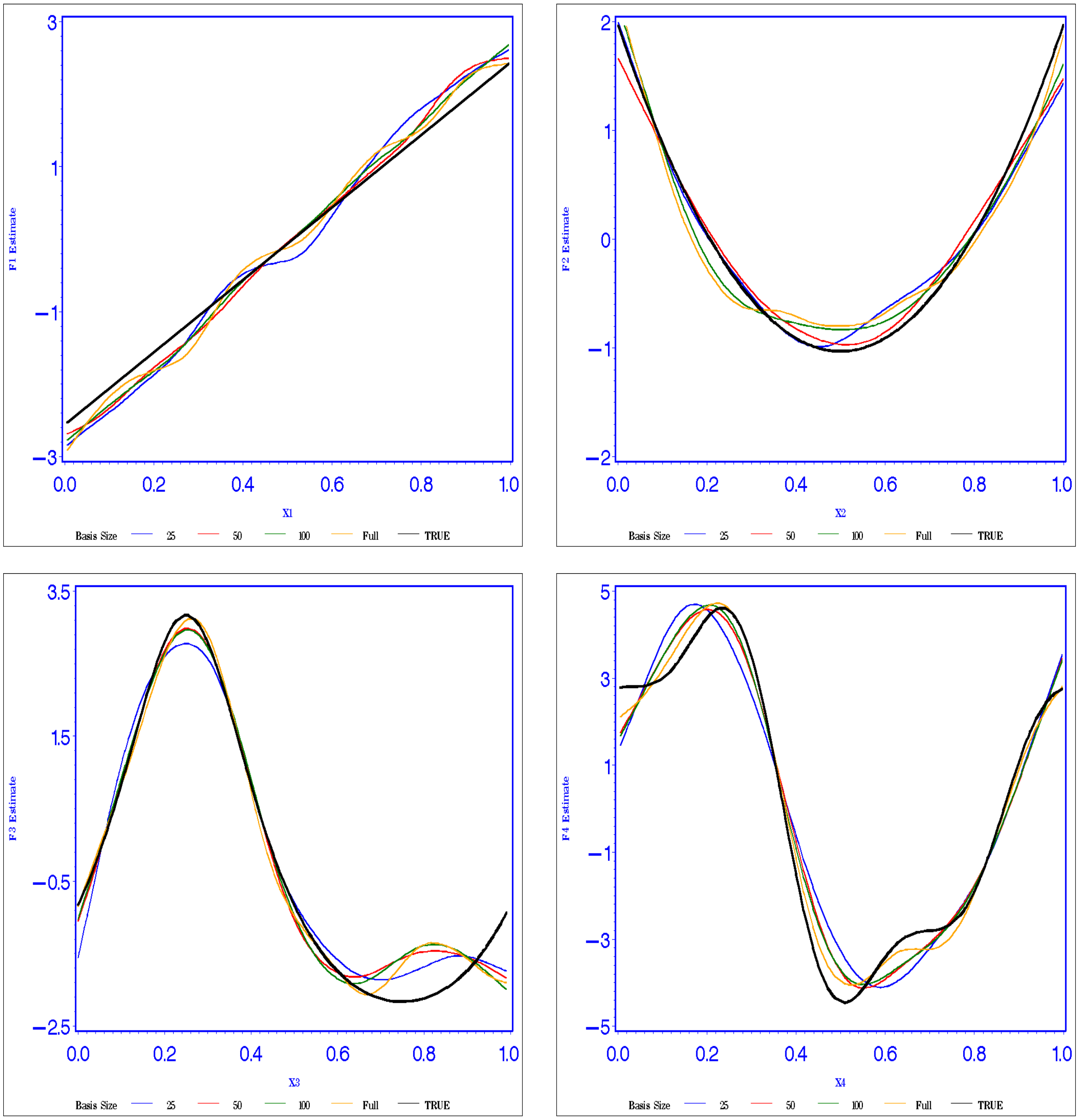

Figure 5.2: Estimated functional components for four explanatory variables in SBA Example.

Notes: This figure shows the component estimates for basis sizes 25, 50, 100 and the full basis Correlated COSSO method along with the TRUE functional components. The plots show the estimates from an additive model applied to the subset basis algorithm for one dataset in SBA simulation, where error standard deviation is $\sigma = 2$ and the sample size is $n = 200$. These four explanatory variables are included in the final model by all four algorithms. For all other variables, the true and estimated functional components are zero.

to the true parameter values. The correct model selection probabilities are in the range of $20-24\%$, and the ISE values are ranging between $2.29-2.44$. So the subset basis algorithm results are close to the full basis algorithm results. This can tell us that the SBA approximates the full basis algorithm consistently well even for a harder component selection problem.

Table 5.9: Subset Basis Algorithm with $n=1000$ Observations ($\sigma=2$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.28(.06)	2.03(.05)	.16(.04)	100	6.00	.00	0:01:02
SBA-CL	25	.29(.04)	2.04(.05)	.20(.05)	100	6.00	.00	0:01:04
SBA-SRS	50	.30(.04)	2.02(.05)	.12(.03)	100	6.00	.00	0:01:05
SBA-CL	50	.29(.04)	2.02(.05)	.15(.04)	100	6.00	.00	0:01:06
SBA-SRS	100	.30(.04)	2.02(.05)	.12(.03)	100	6.00	.00	0:01:52
SBA-CL	100	.30(.04)	2.02(.05)	.13(.03)	100	6.00	.00	0:01:52
SBA-SRS	200	.30(.04)	2.02(.05)	.12(.03)	100	6.00	.00	0:02:54
SBA-CL	200	.30(.04)	2.01(.05)	.12(.03)	100	6.00	.00	0:02:54
Full*	1000	.31(.04)	2.05(.10)	.12(.02)	100	6.00	.00	23:16:54

Notes: This table corresponds to the results from the simulation with $n=1000$, $\rho=0.3$ and $\sigma=2$. The row with * is estimated from only three Monte Carlo simulations because of the extensive computation time. Corresponding SNR is 1.94.

In Tables 5.7 and 5.8 we observe the performance of SBA on simulated datasets with sample size 400. The results from the full basis algorithm and the SBA (especially with basis size 100 or above) are close to each other. For example, the average ISE values with basis size 100 (0.28 for simple random sampling SBA, and 0.30 for cluster SBA) are very close to the average ISE from the full basis algorithm (0.26). The correct model selection probabilities are almost the same. Another important aspect can be gained comparing this table with Table 5.5, which provides the results with the same setting and sample size 200. We observe the improvement of the results in model estimation, variance-covariance parameter estimation and variable selection clearly for both full and subset basis algorithms. This simulation also shows the performance improvement for both the subset and full basis algorithms with the increasing sample size. The same conclusion is reached when the sample size is increased to 1000 in Table 5.9. We would like to remark that all SBA's with any number of basis sizes performed a perfect variable selection performance in this table. In Table 5.10 with the error standard deviation $\sigma = 4$, this performance is again very close to perfect, and the improvement is obvious compared to smaller sample sizes.

The main advantage of the SBA can be observed when the sample size gets larger. We increase the sample size to 400 in Tables 5.7 and 5.8, and to 1000 in Tables 5.9 and 5.10. The gain in computational time is obvious with this larger sample size scenarios. The total computation is approximately $1/10$ of the full basis algorithm for a SBA with basis size 100 for a dataset with 400 observations. The gain is even more for datasets with 1000 observations. The computation time reaches almost a day for a full basis method estimation when the sample size is 1000, yet the SBA with 100 basis size finds an estimate in less than two minutes. This computational burden affects our simulation study, and we only use three Monte Carlo simulations for full basis algorithm when $n = 1000$. This sample size may still not be considered as massive considering the datasets with tens of thousands of observations. For this type of a massive dataset, the full basis algorithm becomes impractical, and even impossible to fit with today's computational resources. Therefore, the SBA is the only feasible algorithm to achieve the approximate Cor-COSSO estimate for massive datasets.

As a short summary of the findings from this simulation example, we observe that the SBA

Table 5.10: Subset Basis Algorithm with $n = 1000$ Observations ($\sigma = 4$).

Method	Basis	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC	Time
SBA-SRS	25	.29(.04)	4.04(.11)	.60(.18)	93	5.99	.06	0:01:12
SBA-CL	25	.30(.03)	4.03(.11)	.52(.17)	94	5.96	.03	0:01:15
SBA-SRS	50	.30(.04)	4.04(.11)	.56(.16)	98	6.00	.02	0:01:06
SBA-CL	50	.30(.04)	4.04(.11)	.52(.16)	97	6.00	.03	0:01:07
SBA-SRS	100	.30(.03)	4.02(.11)	.44(.16)	97	5.99	.02	0:01:54
SBA-CL	100	.30(.03)	4.03(.11)	.46(.14)	96	5.96	.01	0:01:53
SBA-SRS	200	.30(.04)	4.03(.11)	.50(.14)	99	6.00	.01	0:02:54
SBA-CL	200	.30(.04)	4.04(.11)	.52(.17)	96	6.00	.04	0:02:54
Full*	1000	.31(.04)	4.10(.21)	.42(.07)	100	6.00	.00	23:06:09

Notes: This table corresponds to the results from the simulation with $n = 400$, $\rho = 0.3$ and $\sigma = 4$. The row with * is estimated from only three Monte Carlo simulations because of the extensive computation time. Corresponding SNR is 0.97.

method provides a good approximation to the full basis algorithm especially when the sample size gets very large. The comparison of the computational time for these two algorithms reveals the gain of using such an approximation. For very large datasets, the SBA might be the only algorithm which is feasible to find the Cor-COSSO (or Adaptive Cor-COSSO) solution. The prediction accuracy and the variable selection performance results from SBA and the full basis algorithm are comparable. One important issue in SBA is to decide on the basis size to be used. In our experience, we conclude that the basis size of 25 is definitely too small since it did not result in consistent estimates. The performance of SBA with 50 basis functions is also questionable. We recommend using a basis size of at least 100 especially for massive data situations. The approximation with this basis size is satisfactory in our examples, and the computation is not too expensive.

Another aspect of this simulation study is to compare the results from SBA with two sampling methods: Simple Random Sampling (SRS) and Cluster Sampling (CL). From all of six tables, the results from SBA algorithm using SRS and CL sampling methods do not show any significant difference. The expected benefit from using cluster sampling is not achieved. The comparison of both sampling schemes in a higher dimensional problem, where we have a larger number of explanatory variables might be an interesting area of further research. We did not include this kind of a comparison in this dissertation.

5.3 Method Comparisons

We now compare our new proposals: Correlated COSSO and Adaptive Correlated COSSO with an existing method in the literature (original COSSO - Lin and Zhang 2006). The recommendations from the previous chapter is used, i.e., we use L score to tune the smoothing parameter M, and the one-step update algorithm throughout this section. Four simulation examples, featured with different component selection and model estimation scenarios are presented.

The first simulation example compares the methods in a problem containing 10 variables, 4 of which have important effects on the response. We consider $n = 100, 150$ and 200, and scenarios where the explanatory variables are either independent or correlated. The method performance with a stronger correlation among observations is also investigated.

The second example is a larger dimensional problem with 30 explanatory variables, only 8 of which are important. In the third example, we investigated the case, where significant interaction effects are also present in the data generating process.

5.3.1 Example 1: Main Effects Model

To our knowledge, there are very few variable selection methods working with correlated random errors in nonparametric regression literature. In this simulation example, we compare the Correlated COSSO and the Adaptive Correlated COSSO methods with the original COSSO, which ignores the possible correlation among error terms. In Lin and Zhang (2006), it has been shown that the original COSSO method has better variable selection and prediction accuracy performance compared to the popular MARS algorithm (Friedman, 1991).

Table 5.11: Method Comparison in Main Effects - Example 1 ($n = 100$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC
Orig-COSSO	- (-)	2.06(.22)	1.62(.57)	31	5.80(.49)	.56(.50)
Corr-COSSO	.29(.12)	2.00(.20)	1.20(.47)	69	5.87(.54)	.19(.39)
A-C-COSSO	.29(.12)	1.98(.19)	1.09(.45)	76	5.85(.41)	.12(.33)

Notes: This table corresponds to the results from comparison of three methods. The simulated datasets corresponds to the scenario with $n = 100$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.76.

The simulation setting is following. The regression function to generate the relationship is $f(x) = 5g_1(x^{(1)}) + 3g_2(x^{(2)}) + 4g_3(x^{(3)}) + 6g_4(x^{(4)})$, where $g_1, \ldots, g_4$ are defined in Section 5.1. A within-subject compound symmetry (CS) covariance structure is assumed. The correlation matrix corresponding to this simulation study is in the form of W_τ^{-1} in equation (1.19). The within-subject correlation is $\rho = 0.3$ and error standard deviation is $\sigma = 2$. We also allow the number of subjects to increase to show the asymptotic improvement of the results with increasing sample size. Respectively, 20, 30 and 40 subjects are used, which corresponds to sample sizes of 100, 150 and 200 observations per dataset. The SNR corresponding to this example is estimated as $1.76, 1.77$ and 1.80 respectively.

Table 5.12: Method Comparison in Main Effects - Example 1 ($n = 150$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	2.02(.15)	.96(.41)	76	5.95(.22)	.19(.39)
Corr-COSSO	.28(.09)	2.02(.15)	.82(.34)	79	5.93(.26)	.14(.35)
A-C-COSSO	.28(.09)	1.99(.14)	.71(.25)	85	5.86(.43)	.04(.20)

Notes: This table corresponds to the results from comparison of three methods. The simulated datasets corresponds to the scenario with $n = 150$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.77.

Tables 5.11, 5.12 and 5.13 summarize the performance of three methods in three sample size settings. It can be seen that both proposed methods outperform the Original COSSO. Both Adaptive Correlated COSSO and Correlated COSSO methods achieve more accurate predictions (i.e lower ISE's), and better model selection performances (higher correct model selection percentages).

Looking at each of three tables more carefully, we would see that the Adaptive Cor-COSSO performs the best among three methods in terms of both variable selection performance and model estimation accuracy. In all three tables, the smallest ISE's are provided by the Adaptive Cor-COSSO followed by Cor-COSSO. With regard to variable selection, the Adaptive Cor-COSSO also provides the smallest number of incorrect 0's, meaning, this method misses the least number of important variables among the three.

The variance-covariance parameter estimates are overall accurate for both Correlated COSSO

Table 5.13: Method Comparison in Main Effects - Example 1 ($n = 200$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	1.98(.12)	.70(.33)	81	5.91(.35)	.12(.33)
Corr-COSSO	.28(.09)	1.98(.12)	.57(.24)	91	5.96(.20)	.05(.22)
A-C-COSSO	.28(.09)	1.97(.12)	.51(.17)	93	5.93(.29)	.01(.10)

Notes: This table corresponds to the results from comparison of three methods. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.3$ and $\sigma = 2$. Corresponding SNR is 1.80.

and Adaptive Correlated COSSO methods in all three settings. The accuracy of these estimates not only provide information about variance-covariance structure of the data, but they also help improving the model estimation performance.

The three tables (Tables 5.11, 5.12 and 5.13) together represent the well known phenomenon of nonparametric regression that, increasing the sample size will result in better estimates. For all three methods (Original, Correlated, and Adaptive-Correlated COSSO), the better performance of prediction accuracy, variable selection and variance-covariance parameter estimation accuracy can be observed with increasing sample size. For $n = 200$, even the misspecified original COSSO method shows an improved variable selection performance (Table 5.13, $\pi_c = 81\%$). Yet, methods taking the correlation into consideration still works better in variable selection performance ($\pi_c = 91\%$ for Correlated COSSO and $\pi_c = 93\%$ for Adaptive Correlated COSSO). These two methods constantly give smaller ISE values as well.

In Figure 5.3, the estimated functional components are plotted along with the true functional component: Correlated COSSO, Adaptive Correlated COSSO and the original COSSO methods. Notice the components are centered according to the ANOVA decompositions.

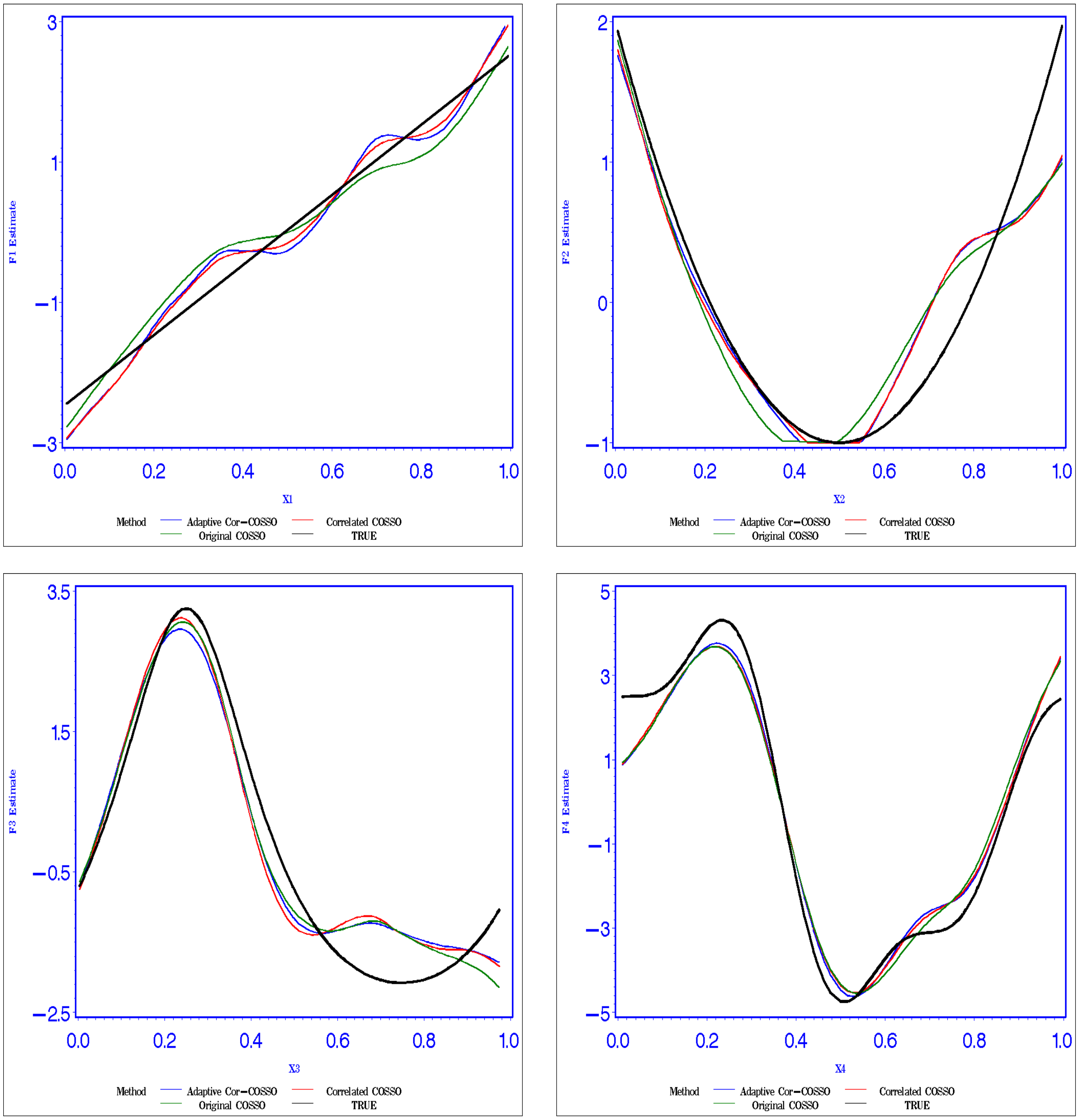

Figure 5.3: Estimated functional components for four explanatory variables in Example 1.

Notes: This figure shows the component estimates from Correlated COSSO, Adaptive Correlated COSSO and original COSSO methods along with the TRUE fit. The plots show the estimates from an additive model applied to the methods in one dataset from simulation Example 1, where error standard deviation is $\sigma = 2$ and the sample size is $n = 200$. These four explanatory variables are included in the final model by all three methods. For all other variables, the true and estimated functional components are zero.

Overall, all three methods provide good functional component approximations. Although it is not easy to compare the fits just by looking at the plots, the Adaptive Correlated COSSO and the Correlated COSSO methods give a very close fit to each other, while the fir from original COSSO shows some divergence for $X1-X3$. On the contrary, for $X4$, all three methods provide close fits. The proposed methods estimate the functional components closely to the true functional component especially for $X1$ and $X2$.

The example above assumes the explanatory variables are *independent* from each other. Yet, this is rarely true for real data. Therefore, the accuracy and variable selection performance of the proposed methods on correlated explanatory variables is also an important aspect. In the following part of this simulation example, we investigate the scenario with correlated explanatory variables.

Tables 5.14 and 5.15 provide results from the simulation study with correlated explanatory variables. Everything about the regression function and errors are the same, but the following method is used to generate the explanatory variables:

- Generate $w_1,\dots,w_{10}$ and u independently from Uniform(0,1)
- Define $x^{(j)} = (w_j + tu)/(1+t)\ for j = 1,\dots,10$, for some $t \geq 0$.

Then the generated variables have $corr(x^{(j)},x^{(k)}) = t^2/(1+t^2)$. We used $t = 0,1$ and 2, where 0 corresponds to the independent $x^{(j)}$'s are already provided above, and $t = 1$ gives a pairwise 0.5 correlation (see Table 5.14) and $t = 2$ gives 0.8 correlation (see Table 5.15).

In both Tables 5.14 and 5.15 it can be seen that the ordering of the methods by means

Table 5.14: Method Comparison in Main Effects - Correlated Explanatory Variables ($t = 1$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	2.00(.12)	.77(.35)	48	5.85(.39)	.53(.64)
Corr-COSSO	.28(.08)	2.00(.13)	.67(.31)	48	5.87(.48)	.48(.59)
A-C-COSSO	.28(.08)	1.98(.12)	.58(.22)	58	5.81(.26)	.26(.44)

Notes: This table corresponds to the results from comparison of three methods. The simulated datasets corresponds to the scenario with $t = 1$, $n = 200$, $\rho = 0.3$ and $\sigma = 2$. There is a 0.5 correlation in between each pair of explanatory variables. Corresponding SNR value is 1.71.

Table 5.15: Method Comparison in Main Effects - Correlated Explanatory Variables ($t = 2$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	2.02(.14)	.80(.25)	19	5.68(.51)	1.04(.72)
Corr-COSSO	.28(.09)	2.02(.14)	.73(.29)	26	5.75(.46)	.96(.75)
A-C-COSSO	.28(.09)	2.01(.12)	.66(.24)	28	5.62(.56)	.73(.62)

Notes: This table corresponds to the results from comparison of three methods. The simulated datasets corresponds to the scenario with $t = 2$, $n = 200$, $\rho = 0.3$ and $\sigma = 2$. There is a 0.8 correlation in between each pair of explanatory variables.Corresponding SNR value is 1.72.

of variable selection and prediction accuracy performance does not change with the correlated explanatory variables. Both Cor-COSSO and Adaptive Cor-COSSO work better than the original COSSO, and the Adaptive Cor-COSSO method performs the best. The pairwise correlation in between explanatory variables makes the component selection problem harder, and this can be seen from these two tables when compared to Table 5.13. Correct model selection percentages decrease to 26% (or 28% for Adaptive Cor-COSSO) when the pairwise correlation is 0.8. Even in this challenging variable selection example, the Correlated COSSO methods perform better than original COSSO.

In this simulation example, we also would like to see the changes in the performance of Cor-COSSO and Adaptive Cor-COSSO methods when the correlation among the error terms are stronger. Two main ideas in this study is to see the limits of the methods by means of how strong a correlation the methods can handle, and to observe how the prediction accuracy and variable selection performance are affected by increased correlation. The within-subject correlation parameter (ρ) is varying. We use $\rho = 0.5, 0.7, 0.8$ and 0.9 for this example.

Table 5.16: Method Comparison in Main Effects - High Correlation Example ($\rho = 0.5$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	1.98(.15)	.72(.36)	78	5.93(.26)	.15(.36)
Corr-COSSO	.48(.08)	1.97(.14)	.45(.16)	98	5.99(.14)	.01(.10)
A-C-COSSO	.49(.08)	1.96(.14)	.41(.15)	97	5.95(.30)	.00(.00)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.5$ and $\sigma = 2$. Corresponding SNR is 1.99.

As the correlation within each subject is increased, we expect the Correlated COSSO and Adaptive Correlated COSSO methods to show better performance, since these methods gain information from this correlation. We expect to see smaller ISE values and larger correct model selection performances. On the other hand, the model estimation might be disturbed when the correlation gets closer to 1 in absolute value. We would like to see how large a correlation can the proposed methods handle with this simulation example.

Table 5.17: Method Comparison in Main Effects - High Correlation Example ($\rho = 0.7$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	1.97(.19)	.72(.36)	81	5.95(.23)	.14(.35)
Corr-COSSO	.69(.06)	1.96(.18)	.33(.13)	100	6.00(.00)	.00(.00)
A-C-COSSO	.70(.06)	1.96(.18)	.31(.13)	99	5.98(.20)	.00(.00)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.7$ and $\sigma = 2$. Corresponding SNR is 2.38.

Tables 5.16 - 5.19 summarizes the results from the simulation study with several correlation parameters ($\rho = 0.5, 0.7, 0.8$ and 0.9 respectively). The increasing signal-to-noise ratio with increasing correlation parameter (ρ) shows that there are more available information in data which the methods taking the correlation into account can use. The larger the correlation, the better the performance of the Cor-COSSO and Adaptive Cor-COSSO methods. Smaller ISE and larger correct model performance can be observed. On the other hand, since the original COSSO method ignores the correlation, no improvement can be observed with increased correlation in terms of ISE. In other words, the gap in between the performances of Cor-COSSO (or Adaptive Cor-COSSO) and original COSSO widens with larger correlation.

Table 5.18: Method Comparison in Main Effects - High Correlation Example ($\rho = 0.8$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	1.96(.21)	.72(.37)	82	5.95(.22)	.13(.34)
Corr-COSSO	.79(.05)	1.96(.19)	.27(.13)	99	5.99(.10)	.00(.00)
A-C-COSSO	.79(.05)	1.96(.19)	.26(.13)	99	5.99(.10)	.00(.00)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.8$ and $\sigma = 2$. Corresponding SNR is 2.77.

The ordering of the method performances are similar. Both Cor-COSSO and Adaptive Cor-COSSO methods outperform the original COSSO, especially with higher correlation. For example, it can be seen in Table 5.19 that with $\rho = 0.9$, both Cor-COSSO and Adaptive Cor-COSSO selects the correct model almost perfectly (99% of the time) while for the original COSSO, the correct model selection probability is only 80%. From the same table, the ISE values are .19 and .20 for Cor-COSSO and Adaptive Cor-COSSO respectively, while this value is .70 for the original COSSO. The main reason is the information gained from the correlation in both Cor-COSSO and Adaptive Cor-COSSO. As long as the correlation gets stronger, the amount of information being used increases, hence the prediction accuracy of the methods gets better. On the other hand, the original COSSO ignores this correlation, therefore does not use this additional information. This results in less efficient results. It can be observed that with increased correlation, ISE values do not decrease for the original COSSO, but the gain is obvious for the methods using the correlation among observations.

Table 5.19: Method Comparison in Main Effects - High Correlation Example ($\rho = 0.9$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	1.96(.23)	.73(.40)	80	5.94(.24)	.14(.35)
Corr-COSSO	.90(.02)	1.95(.21)	.20(.14)	99	5.99(.10)	.00(.00)
A-C-COSSO	.89(.02)	1.96(.21)	.19(.14)	99	5.99(.10)	.00(.00)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.9$ and $\sigma = 2$. Corresponding SNR is 3.62.

In order to see the limits of the Cor-COSSO and Adaptive Cor-COSSO methods, we used even higher correlation in data generation. The convergence was not an issue in either Cor-COSSO or Adaptive Cor-COSSO until the correlation parameter is $\rho = 0.97$. The Correlated COSSO method provide 80% convergence rate at this value of ρ, while Adaptive Correlated COSSO method still provide a 100% convergence. When the correlation is $\rho = 0.99$, the Cor-COSSO convergence rate goes down to 50%, while this rate is still 95% for Adaptive Cor-COSSO. The convergence rates are still quite high for the methods even though the correlation parameter is very high. Especially for the Adaptive Correlated COSSO method, there is almost no convergence issue, even when the within-subject correlation is almost 1.

5.3.2 Example 2: Main Effects Model in Large Dimensions

We consider a larger additive model with 30 variables in this example. The underlying regression function is:

$$f(x) = g_1(x^{(1)}) + g_2(x^{(2)}) + g_3(x^{(3)}) + g_4(x^{(4)}) + 2g_1(x^{(5)}) + 2g_2(x^{(6)}) + 2g_3(x^{(7)}) + 2g_4(x^{(8)}).$$

There are 22 uninformative variables, which makes the example a more sparse variable selection problem. We implement the same compound symmetry correlation structure of Section 5.3.1 with $\sigma = 0.5$ and within-subject correlation $\rho = 0.3$. We used 40 subjects with 5 observations per subject, a total of 200 observations. The estimated signal-to-noise ratio for this example is 7.20 for $\sigma = 0.5$ and 3.60 for $\sigma = 1$.

Table 5.20: Method Comparison in Large Dimensional Variable Selection ($\sigma = 0.5$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR(se)	INC(se)
Orig-COSSO	- (-)	.63(.08)	.31(.13)	.02	19.16(1.39)	1.30(.61)
Corr-COSSO	.30(.09)	.50(.04)	.09(.03)	.54	21.41(.86)	.07(.26)
A-C-COSSO	.29(.09)	.50(.03)	.08(.02)	.84	21.84(.37)	.00(.00)

Notes: This table corresponds to the results from the 30 dimensional variable selection example, where only 8 of them provide information on response. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.3$ and $\sigma = 0.5$. Corresponding SNR is 7.20.

This simulation example is intentionally made more challenging in order to see the limits of the two proposed methods. We compare the original COSSO, the Correlated COSSO and the Adaptive Correlated COSSO methods in Table 5.20. We can see the good covariance parameter estimation performance of the Correlated COSSO and the Adaptive Correlated COSSO. The underlying covariance parameters are in the confidence intervals set by the GML estimates.

The Adaptive Correlated COSSO fit gives the smallest ISE values among the three methods. The reason is that, in order for both Correlated COSSO and original COSSO methods to shrink the uninformative functional component estimates towards zero, these methods use higher penalties on each term, which results in large bias for important variables. In contrast, the Adaptive Correlated COSSO applies different magnitudes of penalization to different variables. Based on an initial estimate (we use a non-weighted SS-ANOVA fit as an initial estimate, Craven and Wahba 1979), the important variables are penalized less compared to unimportant variables. Therefore, the important variables are shrunk less, which provide closer fit to underlying data generating function. On the other hand, the uninformative variables are penalized more, and they are shrunk to zero faster. This is the main reason why higher correct model selection percentages (π_c) and higher average correct 0's are achieved by the Adaptive Correlated COSSO in Table 5.20. The variable selection performance of Adaptive Correlated COSSO is given in Table 5.20 as well. Higher correct model selection percentages (π_c), higher average Correct 0's (CORR's) and the smaller Incorrect 0's (INC) are the signs of better performance in variable selection.

One important point that should be clear from these three tables is that both the Correlated COSSO and the Adaptive Correlated COSSO methods are powerful enough not to miss too many important variables. The Incorrect 0's (INC's) column is either zero or close to zero in

Table 5.21: Appearance Frequencies of Important Variables in Example 2 ($\sigma = 1$).

Method	x1	x2	x3	x4	x5	x6	x7	x8	INC(se)
Orig-COSSO	42	8	24	99	94	39	100	100	2.94(1.41)
Corr-COSSO	41	14	20	96	99	53	100	100	2.77(1.42)
A-C-COSSO	64	34	62	97	98	61	100	100	1.84(1.19)

Notes: This table corresponds to the results from the 30 dimensional variable selection example, where only 8 of them provide information on response. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.3$ and $\sigma = 1$. Corresponding SNR is 3.60. Table includes the frequency of appearance for the *informative* variables in the model.

Table 5.20. This property of the Correlated COSSO is crucial in practice.

Table 5.22: Appearance Frequencies of Unimportant Variables in Example 2 ($\sigma = 1$).

Method	x9	x10	x12	x14	x21	x25	x29	CORR(se)
Orig-COSSO	13	11	11	21	34	61	14	20.03(1.94)
Corr-COSSO	0	5	8	15	15	41	14	20.73(1.73)
A-C-COSSO	2	2	1	1	13	11	19	21.05(1.60)

Notes: This table corresponds to the results from the 30 dimensional variable selection example, where only 8 of them provide information on response. The simulated datasets corresponds to the scenario with $n = 200$, $\rho = 0.3$ and $\sigma = 1$. Corresponding SNR is 3.60. Table includes the frequency of appearance for the most frequent *uninformative* variables in the model. The whole list is not provided because of space concerns.

To explore the properties of Correlated COSSO methods on even more challenging problems, we increase the error standard deviation σ to 1. This model corresponds to an estimated SNR of 3.60. We keep the regression function and error covariance structure intact. Because of the very hard nature of this problem, none of the methods was able to select the correct model above 3% of the time. We instead present in Table 5.21 and Table 5.22 the total frequencies of informative and uninformative variables respectively to be selected in the model in 100 runs. Reader should be aware that variables $x^{(1)} - x^{(8)}$ are informative, and therefore a good model selection method should have a higher frequency of these variables compared to the remaining 22 variables.

Table 5.21 provides the frequencies for informative variables, along with the average number of incorrect 0's. The Adaptive Correlated COSSO method includes almost all important variables with the highest frequency. Compared to original COSSO, Correlated COSSO method performs better for variables $x^{(2)}, x^{(5)}$ and $x^{(6)}$, however the improvement gained by this method is not very substantial. The smallest average number of incorrectly excluded variables (1.84) also show that the Adaptive Correlated COSSO is the best method among the three for not losing too many important variables from the model.

In Table 5.22, we present the variable selection frequencies of uninformative variables, and the average number of noise variables kept in the model. Because of the space constraints, we could not provide the frequencies of all 22 noise variables, instead we provide only the most frequently selected ones. The uninformative variable included most frequently by Adaptive Correlated COSSO method is $x^{(29)}$ (with frequency of 19). On the other hand, original COSSO included $x^{(14)}, x^{(21)}$ and $x^{(25)}$ with very high frequencies. More than half of the time, the model selected by original COSSO method includes variable $x^{(25)}$, which does not carry any information on the regression process at all. This is an important sign to show that misleading results may be obtained when ignoring the correlation in data. Correlated COSSO method works better than original COSSO in excluding noise variables. However, Adaptive Correlated COSSO is the best method with the highest average number of correctly excluded noise variables (21.06). The main reason for Adaptive Correlated COSSO to perform the best is its nature of adaptive penalization according to the relative importance of the components.

5.3.3 Example 3: Main and Interaction Effects Model

One important feature of both Correlated COSSO and Adaptive Correlated COSSO methods is their ability to estimate non-additive models as well. As mentioned in Section 1.3.2, the functional ANOVA decomposition can be built to include interaction components. This feature is an advantage of these methods compared to those methods working only with additive structure only.

In this simulation example, we investigate the performance of Cor-COSSO and Adaptive Cor-COSSO methods by including interaction terms in the model. The regression function used to generate the simulated dataset is:

$$f(x) = 6g_1(x^{(1)}) + 4g_2(x^{(2)}) + 3g_3(x^{(3)}) + 3g_1(x^{(1)}x^{(2)}) + 4g_2(\frac{x^{(1)} + x^{(3)}}{2}).$$

The functions g_1, g_2 and g_3 are already defined in the introduction section. The dataset includes 7 exploratory variables: $x^{(1)}, \ldots, x^{(7)}$, which have a Uniform(0,1) distribution. We considered four scenarios in this example, where in two of them the explanatory variables are independent, and in the remaining two, these variables are pairwise correlated with 0.5 correlation. The correlated explanatory variables are generated using the method described in Section 5.3.1. Each scenario is considered with error standard deviation $\sigma = 0.5$ and $\sigma = 1$. The error terms are also assumed to have a within-subject compound symmetry (CS) correlation structure with the correlation parameter $\rho = 0.3$. Each subject has five observations, and totally there are 300 observations. The correlation matrix corresponding to this simulation study is in the form of W_τ^{-1} matrix of equation (1.19). Please refer to Section 1.5 for details about this correlation structure.

We start component selection from the full model which includes all main effects and first order interaction terms. This full model has 7 main and 21 first order interaction terms. Only 5 of these components carry information on response, hence a total of 23 components are noise.

Table 5.23: Method Comparison with Interaction Effects ($\sigma = 0.5$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC
Orig-COSSO	- (-)	.49(.02)	.039(.013)	68	22.96(.24)	.29(.46)
Corr-COSSO	.32(.07)	.49(.02)	.033(.010)	70	22.98(.14)	.28(.45)
A-C-COSSO	.32(.07)	.49(.02)	.026(.007)	90	22.90(.29)	.02(.24)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO when there are interaction effects in the data. The simulated datasets corresponds to the scenario with $n = 300$, $\rho = 0.3$ and $\sigma = 0.5$ with independent explanatory variables. Corresponding SNR value is 6.09.

Table 5.23 summarizes the results from the setting with $\sigma = 0.5$ and $t = 0$ (independent explanatory variables). Similar results with the large dimensional variable selection problem of Section 5.3.2 can be observed. Since most of the components are noise in this example, the improvement gained by the Adaptive Correlated COSSO method is substantial. Although the Correlated COSSO performs better than the original COSSO, the Adaptive Correlated COSSO outperforms both methods, showing 90% correct model selection percentage and an ISE value of 0.26. The ISE values for the other methods are higher compared to Adaptive Cor-COSSO (.32 for Cor-COSSO and .49 for original COSSO).

Table 5.24: Method Comparison with Interaction Effects ($\sigma = 1$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC
Orig-COSSO	- (-)	.99(.04)	.117(.035)	54	22.91(.32)	.38(.49)
Corr-COSSO	.31(.07)	.99(.05)	.101(.032)	52	22.99(.10)	.48(.48)
A-C-COSSO	.32(.07)	.99(.05)	.089(.032)	64	22.93(.26)	.34(.50)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO when there are interaction effects in the data. The simulated datasets corresponds to the scenario with $n = 300$, $\rho = 0.3$ and $\sigma = 1$ with independent explanatory variables. Corresponding SNR value is 3.04.

We observe a similar pattern when the error standard deviation is increased to $\sigma = 1$ (Table 5.24). The Adaptive Cor-COSSO method is the best among three by means of prediction accuracy ($ISE = 0.089$) and component selection ($\pi_c = 62\%$) performance. Although the correct model selection percentage of original COSSO is slightly better compared to Cor-COSSO, the ISE values reveal that the estimation accuracy is better for the latter method.

Table 5.25: Method Comparison with Interaction Effects - Correlated Explanatory Variables ($\sigma = 0.5$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC
Orig-COSSO	- (-)	.50(.02)	.028(.009)	3	23.00(.00)	1.51(.56)
Corr-COSSO	.31(.07)	.50(.03)	.024(.006)	7	23.00(.00)	1.54(.63)
A-C-COSSO	.31(.07)	.50(.02)	.020(.006)	12	22.98(.14)	1.14(.60)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO when there are interaction effects in the data. The simulated datasets corresponds to the scenario with $n = 300$, $\rho = 0.3$ and $\sigma = 0.5$ with 0.5 pairwise correlation between explanatory variables. Corresponding SNR value is 3.98.

When we introduce the pairwise correlation between explanatory variables, the problem becomes much harder, so the component selection performances of all the methods deteriorated significantly. Even the best method Adaptive Cor-COSSO could select the correct model only 12% of the time (see Table 5.25). When the error standard deviation is $\sigma = 1$, this percentage goes further down to 9% for the Adaptive Cor-COSSO (see Table 5.26).

Table 5.26: Method Comparison with Interaction Effects - Correlated Explanatory Variables ($\sigma = 1$).

Method	$\hat{\rho}(se)$	$\hat{\sigma}(se)$	ISE(se)	π_c	CORR	INC
Orig-COSSO	- (-)	1.00(.05)	.090(.033)	1	22.96(.20)	1.67(.49)
Corr-COSSO	.31(.07)	1.00(.05)	.077(.029)	4	22.95(.26)	1.46(.58)
A-C-COSSO	.31(.07)	.99(.05)	.067(.024)	9	22.93(.26)	1.37(.65)

Notes: This table corresponds to the results from comparison of three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO when there are interaction effects in the data. The simulated datasets corresponds to the scenario with $n = 300$, $\rho = 0.3$ and $\sigma = 1$ with 0.5 pairwise correlation between explanatory variables. Corresponding SNR value is 1.99.

Parallel conclusions can be drawn from Tables 5.25 and 5.26 as well. In this challenging component selection problem, the Correlated COSSO method works slightly better than original COSSO, yet the method using adaptive weights clearly outperforms the other two. Similar conclusions are drawn in the large dimensional variable selection problem. We conclude that especially when most of the components in the model are noise, the Adaptive Correlated COSSO method is a better choice for component selection and model estimation.

CHAPTER 6

REAL EXAMPLES

6.1 Introduction

The main purpose of this section is to illustrate the implementation of Cor-COSSO and Adaptive Cor-COSSO methods to real datasets. We apply both Correlated COSSO and Adaptive Correlated COSSO methods to two real datasets. To show the diverse areas of application, we consider datasets from two separate disciplines, an environmental study data and a dataset which is related to economy. Two datasets are the Ozone data (Breiman and Friedman, 1985; Buja, Hastie, and Tibshirani, 1989; Breiman, 1995; Lin and Zhang, 2006), and the Money Demand data (from SAS/ETS User Guide).

6.2 Real Example 1: Ozone Data

The first example we use to illustrate the Correlated COSSO and Adaptive Correlated COSSO methods is the Ozone Data (Breiman and Friedman, 1985; Buja, Hastie, and Tibshirani, 1989; Breiman, 1995; Lin and Zhang, 2006). The data consists of daily maximum ozone readings and 8 meteorological variables recorded in Los Angeles basin for 330 consecutive days. The response variable (OZONE) is the daily maximum one-hour-average ozone readings. The 8 explanatory variables are TEMP - Temperature (degrees F) measured at El Monte CA, INVHT - Inversion base height (feet) at LAX, PRES - 500 millibar pressure height (m) measured at Vandenberg AFB, VIS - Visibility (miles) measured at LAX, HGT - Pressure gradient (mm Hg) from LAX to Daggett CA, HUM - Humidity (%) at LAX, INVTMP - Inversion base temperature (degrees F) at LAX and WIND - Wind speed (mph) at Los Angeles International Airport (LAX). For more information on the dataset, please refer to Breiman and Friedman (1985).

The Ozone dataset is a time series dataset, where every observation is taken in consecutive days. When we plot the data, we observe a possible AR(1) type of relationship. Based on the Durbin-Watson test, the AR(1) correlation structure is appropriate for the dataset, and we decide to use this correlation structure in the analysis of both Cor-COSSO and Adaptive Cor-COSSO. One-Step Update algorithm is used to fit both methods.

We analyze the Ozone data with the additive model. In other words, we started our analysis with the assumption that there are no interaction effects in the data. The magnitudes of the functional components are measured by their empirical L_1 norms. Empirical L_1 norm is defined as $1/n\sum_{i=1}^{n}|\hat{f}_j(x_i^{(j)})|$ for $j = 1,\ldots,8$, and each j corresponding to one of the eight explanatory variables. In Figure 6.1, we plotted the L_1 norms of these component estimates against the tuning parameter. Both Correlated COSSO and Adaptive Correlated COSSO methods select $M = 3$ by L method. The vertical lines in both figures show the resulting L_1 norms.

The variables TEMP, HUM and INVTMP are included in the final model based on this criterion.

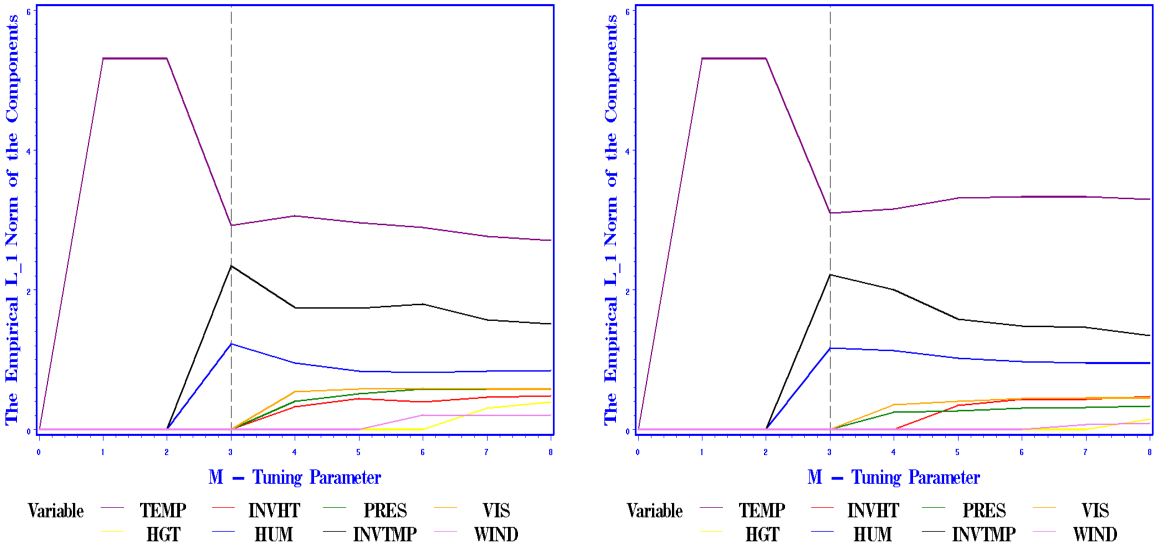

Figure 6.1: The empirical L_1 norm of the estimated components as plotted against the tuning parameter M for Ozone data.

Notes: This figure shows the tuning plots for both Correlated COSSO and Adaptive Correlated COSSO methods applied to the Ozone data. The left panel corresponds to the tuning plot for Cor-COSSO while the right panel corresponds to the Adaptive Cor-COSSO. Both methods select $M = 3$ using the L method.

As mentioned above, both Cor-COSSO and Adaptive Cor-COSSO methods include three variables in the final model. On the other hand, when we apply original COSSO to this dataset, we observed that the method included two more variables in addition to TEMP, HUM and INVTMP. Table 6.1 provides the estimated L_1 norms for the final estimates of three methods.

The variance covariance parameter estimates from the three methods are provided in Table 6.2. Since the original COSSO assumes the observations are independent, this method does not estimate an AR(1) correlation parameter.

Table 6.1: Empirical L_1 Norms of the Estimated Components in Ozone Data.

Method	TEMP	INVHT	PRES	VIS	HGT	HUM	INVTMP	WIND
Orig-COSSO	3.41	1.02	0.18	0	0	1.30	1.18	0
Corr-COSSO	2.92	0	0	0	0	1.23	2.34	0
A-C-COSSO	3.10	0	0	0	0	1.17	2.21	0

Notes: This table presents the empirical L_1 norms of the estimated components as a measure of magnitude of importance for the corresponding component. The results contain output from three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO.

Table 6.2: Variance Component Estimates for Ozone Data.

Method	$\hat{\sigma}$	$\hat{\rho}$
Orig-COSSO	4.23	-
Corr-COSSO	4.31	.29
A-C-COSSO	4.31	.29

Notes: This table presents the variance-covariance parameter estimates from Original COSSO, Correlated COSSO and Adaptive Correlated COSSO fits for the Ozone Data Example.

Figure 6.2 contains the estimated functional components for the Ozone dataset. The plot shows the estimates for variables Temperature (TEMP), Humidity (HUM) and INVTMP, since all other five components are excluded from the final model. The estimated components from Cor-COSSO and Adaptive Cor-COSSO methods are plotted along with the original COSSO estimates of these components.

In order to compare the estimates form Cor-COSSO, Adaptive Cor-COSSO and original COSSO, we estimate the Prediction Squared Error (hereafter PSE) using the following approach. We use two different cross validation sets. First, we use the last 33 observations (one-tenth of the sample size) as the test set (PSE-1), and use the remaining as the training set. Second, the first 33 observations are used as the test set (PSE-2). The reason for this

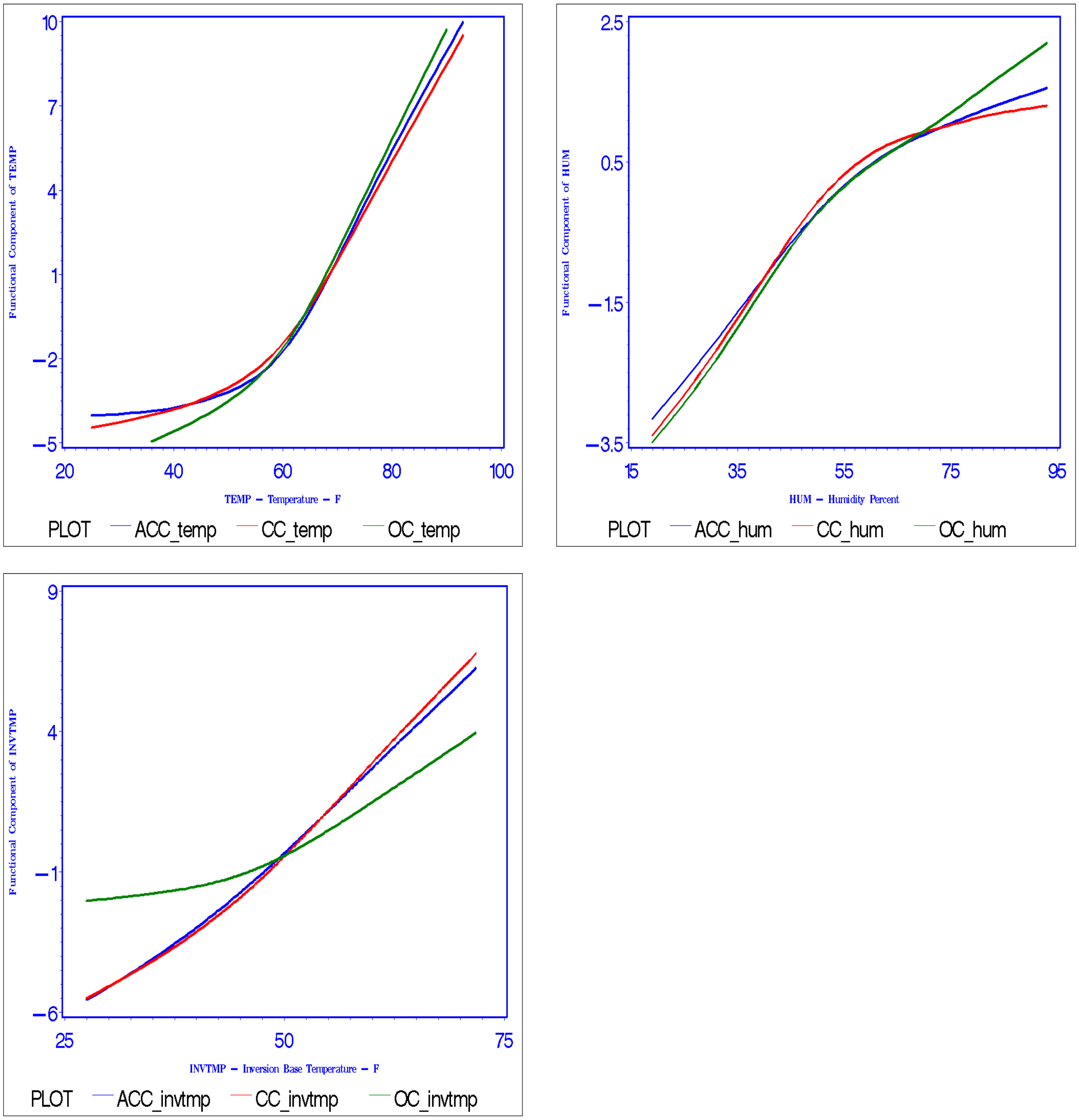

Figure 6.2: Estimated functional components for significant explanatory variables in the Ozone Data Example.

Notes: This figure shows the component estimates for Correlated COSSO, Adaptive Correlated COSSO and original COSSO methods. The plots show the estimates from an additive model applied to the Ozone data. Three components TEMP, HUM and INVTMP are included in the final model by both Cor-COSSO and Adaptive Cor-COSSO methods.

approach is that, if we randomly sample one-tenth of the observations to create the test set, the correlation structure in the dataset would be disturbed. Therefore, we used the consecutive 297 observations in order to maintain the time series structure. The PSE estimates from both cross-validation sets is provided in Table 6.3.

Table 6.3: Predicted Squared Error (PSE) Estimates for Ozone Data.

Method	PSE-1	PSE-2
Orig-COSSO	11.94	11.65
Corr-COSSO	11.77	11.25
A-C-COSSO	11.63	10.81

Notes: This table presents the PSE estimates from Original COSSO, Correlated COSSO and Adaptive Correlated COSSO fits. The smallest PSE estimates are provided by the Adaptive Correlated COSSO method. The PSE-1 is estimated using the last 33 observations, while the PSE-2 is estimated using the first 33.

From Table 6.3, the Adaptive Cor-COSSO method has the smallest PSE values. Although we do not have the standard error estimates for the PSE's, and therefore cannot make a valid comparison, the two proposed methods show better results by means of the prediction accuracy.

6.3 Real Example 2: Money Demand Data

In this section, we apply the proposed Correlated COSSO and Adaptive Correlated COSSO methods to another time series dataset. The analysis is intended to illustrate our methods rather than providing a formal analysis to the dataset at hand. We used Money Demand dataset which is already analyzed in SAS/ETS User's Guide. The purpose is to model the log-log demand with four explanatory variables. The response is real money stock (M). Explanatory variables are lagged response divided by current Gross Domestic Product (M1CP), real Gross National Product (Y), yield on corporate bounds (INTR) and rate of prices changes (INFR). All variables are log transformed prior to the analysis. Although we do not need transformations in nonparametric regression, in order to show the model misspecifications in linear model, we used the transformed variables that are already used in analyzing this dataset. Data contains observations from 1968-second quarter to 1983-fourth quarter. There are 63 observations in the dataset. Please refer to Balke and Gordon (1986) for more details on the dataset.

A linear model is applied to the Money Demand data in SAS/ETS user guide example. The model shows that an first-order Autoregressive (AR-1) error covariance structure is suitable for the dataset. This conclusion is reached after comparison with a Durbin-Watson test. In the final model, all four variables are assumed to have a linear relationship with the response (M1).

6.3.1 Additive Model for Demand Data

First, we consider an additive model which assumes no interaction effects exists in the dataset. We applied both Correlated COSSO and Adaptive Correlated methods using this additive model as the full model. One-Step Update algorithm is used to fit both methods. The L method includes all four variables into the final model for both Cor-COSSO and Adaptive Cor-COSSO. Therefore, none of the components are excluded.

The Correlated COSSO method estimated the error standard deviation (σ) as 0.02, and a positive auto-correlation as $\rho = 0.96$. The variance covariance parameter estimates from the Adaptive Correlated COSSO estimates are the same with Cor-COSSO. The estimated functional components for these two methods and the original COSSO fit are plotted against the explanatory variables in Figure 6.3.

As it can be seen from the Figure 6.3, although for some of the variables the linearity assumption can be made, especially for INFR variable, the relationship is far from being linear. The form of the relationship in between this variable and the response (M1) can be misleading, and even may be lost with the linearity assumption in a variable selection problem. Due to the nonparametric estimation, the Correlated COSSO method does not assume any specific form for this relationship, hence the method captures nonlinear information as well.

6.3.2 Interaction Model for Demand Data

Now, we consider a model which takes the two-way interaction terms into account for the Money Demand dataset. We apply both Correlated COSSO and Adaptive Correlated methods to the two-way interaction model. One-Step Update algorithm is used to fit both methods.

Figure 6.4 shows how the magnitudes of the estimated components change with M. The left panel shows the tuning plot for Cor-COSSO while right panel is the same plot for the Adaptive Cor-COSSO. These magnitudes are measured by their empirical L_1 norms, defined by $1/n\sum_{i=1}^{n}|\hat{f}_j(x_i^{(j)})|$ for $j = 1, \ldots, 10$, and each j corresponding to one of the four explanatory

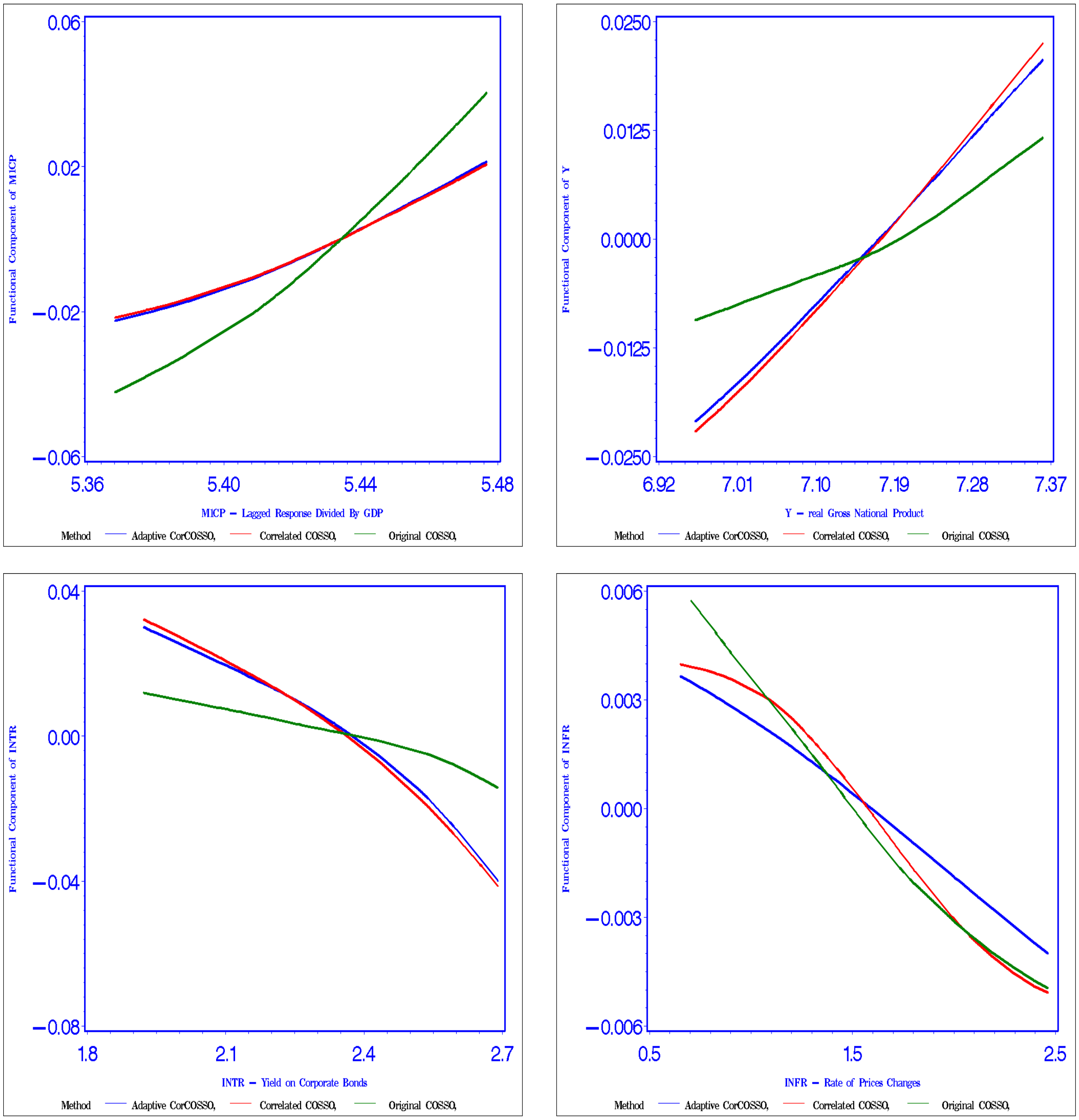

Figure 6.3: Estimated functional components for each of four explanatory variables in additive Money Demand Model.

Notes: This figure shows the component estimates for Correlated COSSO, Adaptive Correlated COSSO and original COSSO methods. The plots show the estimates from an additive model applied to the Money Demand data. All four components (explanatory variables) are included in the final model by all three methods.

variables. L method (Diggle and Hutchinson 1989) is used for tuning M as recommended by the simulation studies. Both methods choose $M = 4$ for this data set.

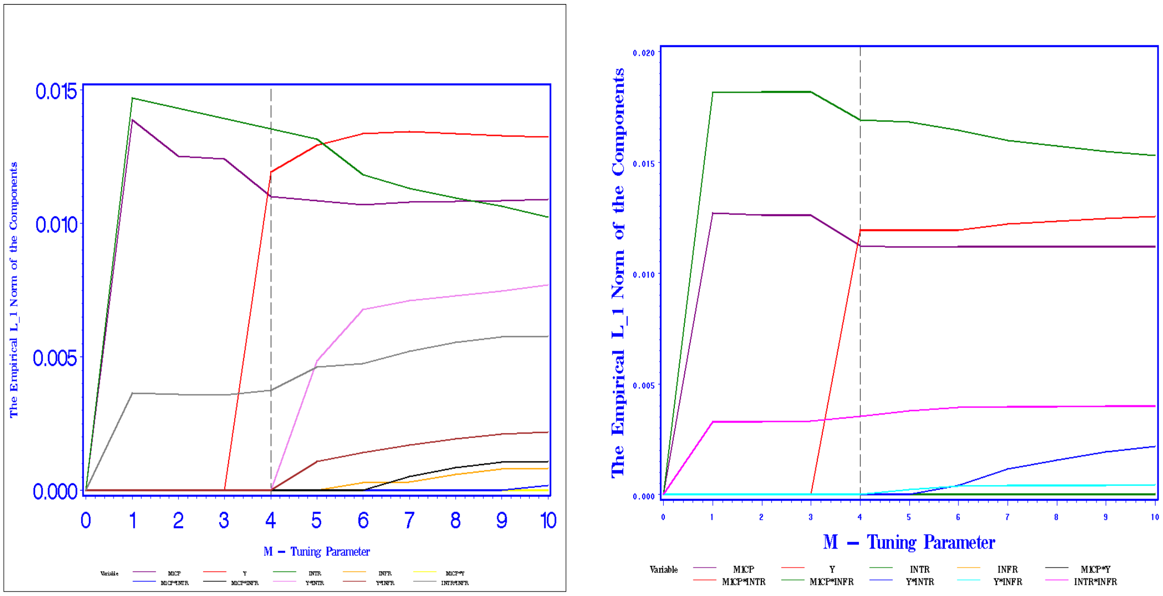

Figure 6.4: The empirical L_1 norm of the estimated components as plotted against the tuning parameter M for Money Demand data with two-way interaction model.

Notes: This figure shows the tuning plots for both Correlated COSSO and Adaptive Correlated COSSO methods applied to the Money Demand data with the two-way interaction model. The left panel corresponds to the tuning plot for Cor-COSSO while the right panel corresponds to the Adaptive Cor-COSSO. Both methods select $M = 4$ using the L method.

The estimated functional components for these two methods and the original COSSO fit are plotted against the explanatory variables in Figure 6.5. In this figure, we include only significant main effect terms found by Cor-COSSO and Adaptive Cor-COSSO methods which are Gross Domestic Product (M1CP), real Gross National Product (Y) and yield on corporate bounds (INTR).

There are four components included into the final model for both Cor-COSSO and Adaptive Cor-COSSO. These components are the main effects of M1CP, Y, INTR and the interac-

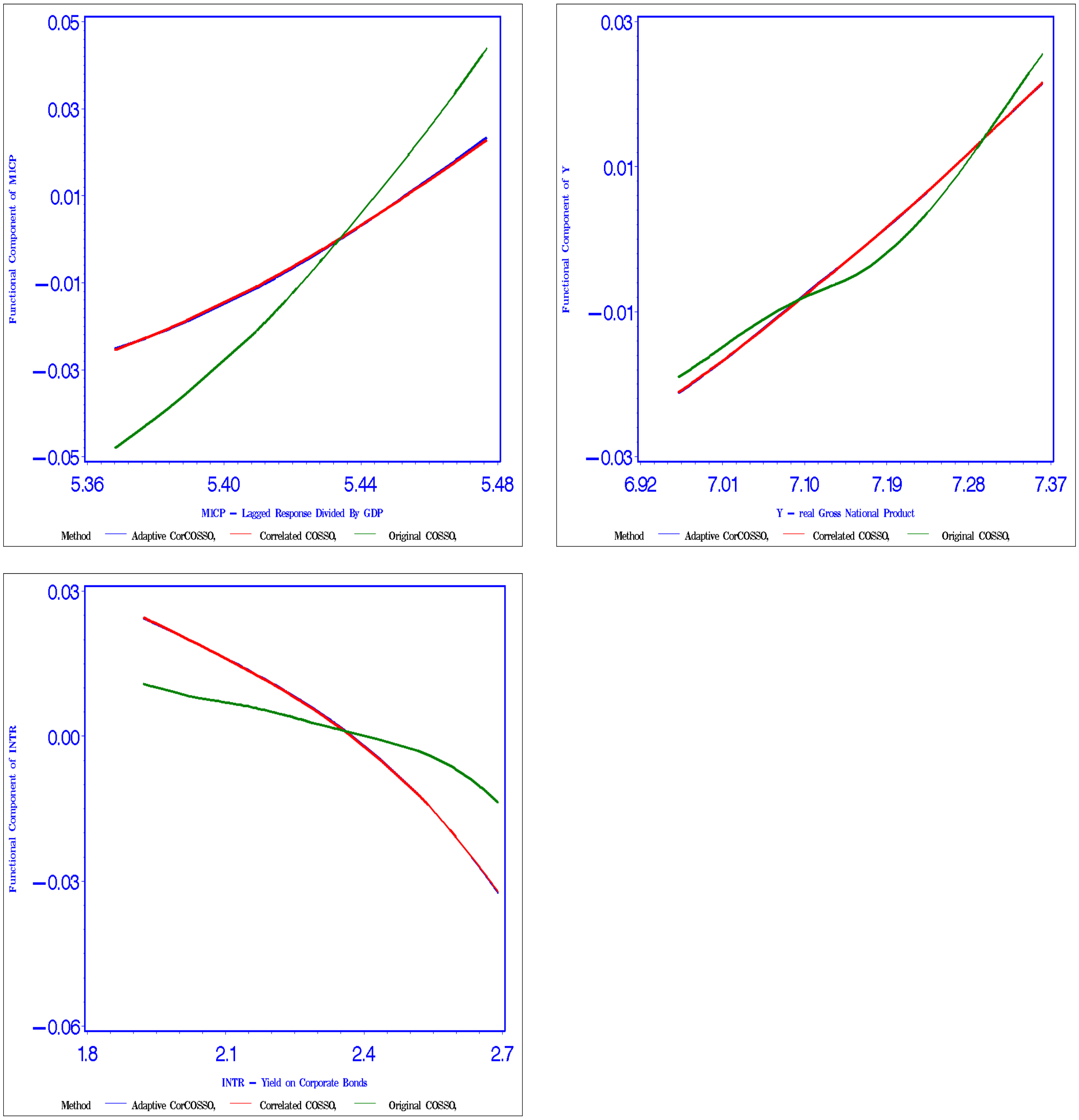

Figure 6.5: Estimated functional components for significant main effects in the two-way interaction Money Demand model.

Notes: This figure shows the component estimates for Correlated COSSO, Adaptive Correlated COSSO and original COSSO methods applied to the Money Demand data with the two-way interaction model. Only significant main effects are plotted. These explanatory variables are M1CP, Y and INTR.

tion term between INTR and the rate of prices changes (INFR). Table 6.4 presents the magnitudes of the estimated functional components for Cor-COSSO, Adaptive Cor-COSSO and original COSSO models.

Table 6.4: Empirical L_1 Norms of the Estimated Components in Money Demand Data with Two-way Interaction Model.

Component	Orig-COSSO	Cor-COSSO	A-C-COSSO
M1CP	2.10	1.10	1.10
Y	1.29	1.19	1.19
INTR	0.76	1.68	1.69
INFR	0.52	0	0
M1CP*Y	0.53	0	0
M1CP*INTR	0	0	0
M1CP*INFR	0.33	0	0
Y*INTR	0.26	0	0
Y*INFR	0	0	0
INTR*INFR	0.74	0.38	0.35

Notes: This table presents the empirical L_1 norms of the estimated components as a measure of magnitude of importance for the corresponding component. The results contain output from three methods: Original COSSO, Correlated COSSO and Adaptive Correlated COSSO. The results are in 10^{-2} scale.

As it can be seen from Table 6.4, the original COSSO method includes several additional main effects and interaction terms to the final model. The comparison of the methods are done based on the Prediction Squared Error estimates which are estimated using the last 10 observation (PSE-1) and the first 10 observations (PSE-2) as the test set. The results can be found in Table 6.5.

One should keep in mind that the Money Demand dataset originally has 63 observations. When we use 10 observations as the test set, the total number of observations left for the training set is only 53, which is quite small for a large non-additive nonparametric model with

Table 6.5: Predicted Squared Error (PSE) Estimates for Money Demand Data with Two-way Interaction Model.

Method	PSE-1	PSE-2
Orig-COSSO	6.14	18.70
Corr-COSSO	5.15	19.12
A-C-COSSO	3.20	20.80

Notes: This table presents the PSE estimates from Original COSSO, Correlated COSSO and Adaptive Correlated COSSO fits. The PSE-1 is estimated using the last 10 observations, while the PSE-2 is estimated using the first 10. All PSE values are in 10^{-4} scale.

10 components. Therefore, the PSE results are not too stable, and show high variations as can be seen from the table. Although no clear conclusions can be reached by looking at the limited results from PSE's, we still believe that in a time series dataset such as Money Demand data, this strong correlation in between the consecutive observations should not be neglected.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this dissertation research, we consider the multivariate regression and the associated variable selection problem for correlated data. The variable selection methods for both linear and nonparametric regression models for independent data are reviewed. We have demonstrated the importance of modelling the correlation among data, which is often ignored in variable selection procedures.

Variable selection in nonparametric regression is a difficult task, which becomes even more challenging for correlated data such as clustered, longitudinal data or repeated measurements. Little work has been done on variable selection for nonparametric models with correlated random errors. In the framework of smoothing spline Analysis of Variance (SS-ANOVA), we propose some unified approaches for simultaneously selecting variables and estimating model parameters and covariance structures. The first new method, as a generalization of the Component Selection and Smoothing Operator (COSSO - Lin and Zhang 2006), imposes a soft-thresholding penalty on functional components for sparse estimation and takes the covariance structure into account at the same time, hence the name Correlated COSSO. The existence of the Cor-COSSO solution is proven and it is shown that the solution has a finite dimensional representation.

In addition, an improvement on the Correlated COSSO method is proposed. The so-called Adaptive-Correlated COSSO method conducts weighted penalization to functional components, achieving improved variable selection and prediction accuracy performances. The weights are assigned to each component in a way that important components are penalized less while uninformative components are penalized more to be shrunk faster to zero. An initial estimate based on the data at hand is necessary to calculate the weights in advance. We propose using a non-weighted smoothing spline ANOVA model as this initial estimate.

We then propose several algorithms to calculate both the Correlated COSSO and the Adaptive Correlated COSSO estimates. These algorithms iterate between a smoothing spline with correlated errors and a quadratic programming. Another important feature of the algorithms is that they allow the use of the available commercial software, such as SAS *Proc Mixed* and *Proc IML*.

The selection of smoothing parameters is very important as it is for any nonparametric regression method. Parameter λ_0 is estimated using the Generalized Maximum Likelihood (GML) approach (Wang, 1998b; Opsomer, Wang, and Yang, 2001). The tuning parameter M, which controls the number of variables to be included in the final model, is crucial for variable selection purposes. Several criteria for choosing M are compared with extensive simulation studies, and the L method (Diggle and Hutchinson, 1989) has shown the best performance.

In spite of the good practical performance, the computation of the Correlated COSSO methods can be timely when the sample size (n) is large. The most time consuming part is the estimation of smoothing spline models with correlated data. A linear mixed effects model representation is used to solve the multivariate smoothing spline problem, which requires the estimation of n random effects in every iteration step of the algorithm. We therefore suggest an alternative subset basis approach for feasible implementation in massive data, which can decrease the number of random effects to be estimated. Our extensive simulation shows that the subset basis algorithm is a good approximation to the full basis algorithm, and the computation time is drastically decreased for large datasets.

The empirical performance of the Correlated COSSO and the Adaptive Correlated COSSO methods are studied and demonstrated using extensive simulations. Our simulation study consists of two parts. The first part is designed to compare different tuning criteria and algorithms for both Cor-COSSO and Adaptive Cor-COSSO. Based on the first set of simulations, we recommend using L method for smoothing parameter (M) selection, and the one-step update algorithm to estimate Cor-COSSO (or Adaptive Cor-COSSO). Moreover, we have investigated the performance of the subset basis algorithm, and conclude the algorithm provides good approximation to the full basis Cor-COSSO when a basis size 100 or more is used. We also compare sampling methods to select the basis functions. Simple random and cluster sampling methods show compatible performance with each other.

The second part of our simulation study gives the comparison of proposed methods, Correlated COSSO and Adaptive Correlated COSSO with the original COSSO as the benchmark method. The comparisons are conducted under a variety of scenarios: correlated and uncorrelated explanatory variables, highly correlated data, a large dimensional variable selection example, and an example containing interaction terms. In these simulation studies we have found that both Correlated COSSO and Adaptive Correlated COSSO methods outperform the original COSSO when there exists correlation in data. Adaptive Correlated COSSO performs superior to the Correlated COSSO especially in larger dimensional variable selection problems. We also observed that both methods' performance are satisfactory when there are interaction effects in the model.

We then apply the proposed methods to two real datasets: the Ozone data (Breiman and Friedman, 1985) and the Money Demand data (from SAS/ETS User Guide). The performance of the methods are compared based on the Predicted Squared Errors (PSE) estimates. Both Cor-COSSO and Adaptive Cor-COSSO methods performed at least as good as the original COSSO in both examples. In Ozone data, the Adaptive Correlated COSSO method provides the smallest PSE results.

We are planning to investigate the asymptotic properties of both Correlated COSSO and Adaptive Correlated COSSO methods. We would like to investigate the theory behind these methods, and study the consistency of these methods in following three aspects: functional component estimation, variance components and covariance parameter estimation, and variable selection.

Models for analysis of repeated measures or longitudinal data most of the time suffer from difficulties with modelling the general covariance structure. Especially, when the data is highly unbalanced, we might need random effects to model the correlation structure (Laird and Ware, 1982). We plan to consider the nonparametric mixed effects models with parametric random effects and nonparametric fixed effects. These kind of models can be useful especially in the situations where the correlation structure is not easy to define. Consider the following model

$$y_i = f(\mathbf{x}_i) + \mathbf{z}_i^{\mathrm{T}}\mathbf{b} + \varepsilon_i, \quad i = 1,...,n, \tag{7.1}$$

where regression function $f(\mathbf{x})$ is assumed to be a smooth function modelling the fixed effects of $\mathbf{x}$, and $\mathbf{z}_i^{\mathrm{T}}\mathbf{b}$ are random effects with $\mathbf{b} \sim N(\mathbf{0}, \mathbf{B})$, and $\varepsilon_i \sim N(0, \sigma^2) \quad i = 1, \ldots, n$ are independent of $\mathbf{b}$ and each other. Examples of these kind of models can be found in Wang (1998a); Gu and Ma (2005). We believe these models will give us more power in modelling the covariance structure. Moreover, the parametric random effects can be implemented in mixed model representation that we use in connection with SS-ANOVA fitting. As a future research area, we would like to investigate the variable selection problem using nonparametric mixed effects models.

BIBLIOGRAPHY

Akaike, H. 1973. "Fitting Autoregressive Models for Prediction." *Annals of The Institute of Statistical Mathematics* 21:243–247.

Altman, N. 1990. "Kernel Smoothing of Data with Correlated Errors." *Journal of the American Statistical Association* 85:749–759.

Aronsajn, N. 1950. "Theory of Reproducing Kernels." *Transections of American Mathematical Society* 66:337–404.

Berger, J., and L. Pericchi. 2001. "Objective Bayesian Methods for Model Selection: Introduction and Comparison (with discussion)." *Model Selection (ed. Lahiri), IMS Lecture Notes* 38:135–207.

Breiman, L. 1995. "Better Subset Regression Using the Nonnegative Garrote." *Technometrics* 37:373–384.

Breiman, L., and J. Friedman. 1985. "Estimating Optimal Transformations for Multiple Regression and Correlation." *Journal of the American Statistical Association* 80:580–598.

Buja, A., T. Hastie, and R. Tibshirani. 1989. "Linear Smoothers and Additive Models (with discussion)." *Annals of Statistics* 17:453–555.

Casella, G., and E. Moreno. 2006. "Objective Bayesian Variable Selection." *Journal of the American Statistical Association* 101:157–167.

Chipman, H. 1996. "Bayesian Variable Selection and Related Predictors." *Canadian Journal of Statistics* 24:17–36.

Craven, P., and G. Wahba. 1979. "Smoothing Noisy Data with Spline Function." *Numerical Mathematics* 31:377–403.

Diggle, P., and M. Hutchinson. 1989. "On Spline Smoothing with Autocorrelated Errors." *Australian Journal of Statistics* 31:166–182.

Diggle, P., K. Liang, and S. Zeger. 2002. *Analysis of Longitudinal Data*. Oxford University Press, Oxford.

Drapper, N., and H. Smith. 1998. *Applied Regression Analysis, 3rd edt.*. Wiley, New York.

Eubank, R. 1988. *Spline Smoothing and Nonparametric Regression*. Dekker, New York.

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

Frank, I., and J. Friedman. 1993. "A Statistical View of Some Chemometrics Regression Tools." *Technometrics* 35:109–148.

Friedman, J. 1991. "Multivariate Adaptive Regression Splines." *The Annals of Statistics* 19:1–141.

George, E., and R. McCulloch. 1997. "Approaches for Bayesian Variable Selection." *Statistica Sinica* 7:339–373.

—. 1995. "Stochastic Search Variable Selection."

—. 1993. "Variable Selection via Gibbs Sampling." *Journal of the American Statistical Society* 88:881–889.

Geweke, J. 1996. "Variable Selection and Model Comparison in Regression." *Bayesian Statistics (eds. Bernardo, Berger, Dawid, Smith)* 5:609–620.

Green, P., and B. Silverman. 1988. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman and Hall/CRC, New York.

Gu, C. 1992. "Cross-validating non Gaussian Data." *Journal Computational and Graphical Statistics* 2:169–176.

—. 2002. *Smoothing Spline ANOVA Models*. Springer, New York, Springer Series in Statistics.

Gu, C., and C. Han. 2004. "Optimal Smoothing with Correlated Data." *Technical report, Department of Statistics, Purdue University* -:–.

Gu, C., and P. Ma. 2005. "Optimal Smoothing in Nonparametric Mixed-Effects Models." *The Annals of Statistics* 33(3):1357–1379.

Halmos, P. 1957. *Introduction to Hilbert Space and Theory of Spectral Multiplicity*. Chelsea, New York.

Harville, D. 1977. "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems." *Journal of the American Statistical Association* 72:320–340.

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

Hastie, T., R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer.

Kimeldorf, G., and G. Wahba. 1971. "Some Results on Tchebycheffian Spline Functions." *Journal of Mathematical Analysis and Applications* 33:82–94.

Laird, N., and J. Ware. 1982. "Random-effects models for longitudinal data." *Biometrics* 38(4):963–974.

Lin, Y., and H. Zhang. 2006. "Component Selection and Smoothing in Smoothing Spline Analysis of Variance Models - COSSO." *Annals of Statistics* 34(5):2272–2297.

Miller, A. 1990. *Subset Selection in Regression*. New York - Chapman and Hall.

Opsomer, J., Y. Wang, and Y. Yang. 2001. "Nonparametric regression with correlated errors." *Statistical Science* 16(2):134–153.

Robinson, G. 1991. "That BLUP is a Good Thing: The Estimation of Random Effects." *Statistical Science* 6:15–51.

Ruppert, D., and R. Carroll. 2000. "Spatially-adaptive Penalties for Spline Fitting." *Australian and New Zealand Journal of Statistics* 45:205–223.

Schwarz, G. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6(2):461–464.

Storlie, C., H. Bondell, B. Reich, and H. Zhang. 2007. "The Adaptive COSSO for Nonparametric Surface Estimation and Model Selection." *Submitted to the Annals of Statistics* -:?–?

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

Tibshirani, R. 1996. "Regression Shrinkage and Selection via LASSO." *Journal of Royal Statistical Society, Ser. B* 58:267–288.

Verbeke, G., and G. Molenberghs. 2000. *Linear Mixed Models for Longitudinal Data*. Springer, New York.

Wahba, G. 1985. "A Comparison of GCV and GML for Choosing the Smoothing Parameters in Generalized Spline Problem." *The Annals of Statistics* 13(4):1378–1402.

—. 1990. *Spline Models for Observational Data*. SIAM, Philadelphia. CBMS-NSF Regional Cenference Series in Applied Mathematics.

Wahba, G., Y. Wang, C. Gu, R. Klein, and B. Klein. 1995. "Smoothing Spline ANOVA for exponential families, with application to the WESDR." *Annals of Statistics* 23:1865–1895.

Wang, Y. 1998a. "Mixed Effects Smoothing Spline ANOVA." *Journal of Royal Statistical Society Ser. B* 60:159–174.

—. 1998b. "Smoothing Spline Models with Correlated Random Errors." *Journal of the American Statistical Association* 93:341–348.

Xiang, D., and G. Wahba. 2006. "Approximate Smoothing Splines for Large Data Sets in Binary Case." *Proceedings of ASS Joint Statistical Meetings, Biometrics Section*, pp. 94–98.

Yau, P., R. Kohn, and S. Wood. 2002. “Bayesian Variable Selection and Model Averaging in High Dimensional Multinomial Nonparametric Resreggion.” *Journal of Computational and Graphical Statistics* 12:23–54.

Zhang, H., and Y. Lin. 2006. “Component Selection and Smoothing for Nonparametric Regression in Exponential Families.” *Institute of Statistics Mimeo Series* 2554:1–23.

Zhang, H., and W. Lu. 2007. “Adaptive Lasso for Coxs Proportional Hazards Model.” *Biometrika* -:?–?

Zhang, H., G. Wahba, Y. Lin, M. Voelker, M. Ferris, R. Klein, and B. Klein. 2004. “Variable Selection and Model Building via Likelihood Basis Pursuit.” *Journal of American Statistical Association* 99:659–672.

Zou, H. 2006. “The Adaptive Lasso and Its Oracle Properties.” *Journal of the American Statistical Association* 101(476):1418–1429.

Zou, H., and T. Hastie. 2005. “Regularization and Variable Selection via the Elastic Net.” *Journal of Royal Statistical Society, Ser. B* 67:301–320.