Nonparametric Methods for Big and Complex Datasets under a Reproducing Kernel Hilbert Space Framework

by

Xiaoxiao Sun

(Under the Direction of Ping Ma)

Abstract

Large and complex data have been generated routinely from various sources, for instance, time course biological studies and social media. Classic nonparametric models, such as smoothing spline ANOVA models, are not well equipped to analyze such large and complex data. To overcome these challenges, I propose novel nonparametric methods under a reproducing kernel Hilbert space framework to (1) significantly reduce daunting computational costs of selecting smoothing parameters for smoothing spline ANOVA models; (2) model the data with a functional response and a functional predictor; (3) accurately identify differentially expressed genes in time course RNA-seq data. To validate my proposed methods, I conduct simulation studies and apply the proposed methods to real data studies. In the end, I present derivations and theoretical proofs.

Index words: smoothing spline ANOVA, smoothing parameters selection, optimal smoothing parameters, function-on-function regression, representer theorem, penalized least squares, reproducing kernel Hilbert space, minimax convergence rate, time course RNA-seq, differentially expressed genes

Nonparametric Methods for Big and Complex Datasets under a Reproducing Kernel Hilbert Space Framework

by

Xiaoxiao Sun

B.S., Central University of Finance and Economics, China, 2010

M.S., Central University of Finance and Economics, China, 2013

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2018

Nonparametric Methods for Big and Complex Datasets under a Reproducing Kernel Hilbert Space Framework

by

Xiaoxiao Sun

Major Professors:	Ping Ma
Committee:	Wenxuan Zhong Ying Xu Abhyuday Mandal Robert Schmitz Pang Du

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2018

Acknowledgments

I can not thank enough my advisor Ping Ma, who has generously provided me with assistance, guidance, and support throughout my Ph.D. study. I have been greatly impressed and influenced by his insights, wisdom, and passion in data science. I would like to express my deepest gratitude to him. I would also like to thank my committee members Drs. Wenxuan Zhong, Ying Xu, Abhyuday Mandal, Robert Schmitz, and Pang Du for working with me during this process. In particular, Dr. Wenxuan Zhong has greatly helped me in my study and research. She is a role model for me as a statistician. Drs. Ying Xu, Abhyuday Mandal, and Robert Schmitz have influenced me with their insightful comments and critiques on my research. Dr. Pang Du greatly helped me during the fourth year of my graduate study. He has taught me various tips on scientific writing and theoretical proof.

I am also honored to have the collaboration with Drs. Jun S. Liu, Xiao Wang, Xiaoyu Zhang, Di Wu, Lexiang Ji, Dong-Hyun Kim, and David Dalpiaz. I really enjoyed and learned a lot in the collaboration with them. Special thanks to members of Big Data Analytics Lab who provide many detailed comments and suggestions on my thesis.

Lastly, I would like to thank the National Institutes of Health (R01GM122080 and R01GM113242) and National Science Foundation (DMS-1440037, DMS-1440038, and DMS-1438957), who have provided funding support for the work.

Contents

Chapter 1

Introduction

1.1 Overview of Big Data Issues

I consider a nonparametric model (Gu, 2013; Wahba, 1990) of the form

$$y_i = \eta(x_i) + \epsilon_i, \quad i = 1, \cdots, n, \tag{1.1}$$

where $y_i \in \mathbb{R}$ is the response variable for the ith observation, η is a nonparametric function varying in an infinite dimensional functional space, and $x_i = (x_{i\langle 1\rangle}, \cdots, x_{i\langle d\rangle})$ is a d-dimensional vector of predictors for the ith observation, ϵ_i's are independent and identically distributed random errors with mean zero and unknown variance σ^2. For multivariate function ($d > 1$), the functional ANOVA can be used to decompose the η into summation of functions in orthogonal subspaces. Then, the nonparametric function η can be estimated by minimizing the penalized least squares

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \eta(x_i))^2 + \lambda J(\eta), \tag{1.2}$$

where $J(\eta)$ is a quadratic roughness penalty, and the smoothing parameter λ controls the trade-off between goals of the lack of fit of η and the roughness of η.

The smoothing spline ANOVA (SS-ANOVA) model provides a flexible approach to find the minimizer $\eta_{n,\lambda}$ minimizing (1.2). Since the minimizer of (1.2) is sensitive to the selection of λ, it is crucial to choose an effective and efficient method for the smoothing parameter selection. Numerous computational methods have been proposed. The C_L method (Mallows, 1973) is one of the earliest ones. However, the C_L method is impractical due to its dependence on the unknown σ^2. To circumvent the problem, Craven and Wahba (1978) proposed the generalized cross-validation (GCV) method. They showed that the smoothing parameter estimated by GCV minimized a risk function asymptotically. Although the GCV method obtains a good estimate of λ without prior knowledge of the variance σ^2, it occasionally has an under-smoothing problem. To curb the problem, Kim and Gu (2004) proposed a modified version of GCV by adding a fudge factor. Under the Bayes framework, Wahba (1985) proposed a generalization of the maximum likelihood (GML) estimate for the smoothing parameter. Extensive simulations were performed to illustrate GML provided satisfying estimates. Nonetheless, under certain conditions, the minimizer $\eta_{n,\lambda}$ based on the smoothing parameter chosen by GML can not attain the optimal convergence rate. Different from the above methods, Hurvich et al. (1998) proposed an improved AIC (Akaike information criterion) method for choosing the smoothing parameter in any linear smoother. Besides the versatility in applying in any linear smoother, the criterion aims to refrain from the under-smoothing problem in GCV. However, the performance of the criterion is not as good as that of other criteria, such as GCV, in some situations (Aydin et al., 2013). Moreover, its theoretical soundness is hard to justify due to the lack of theoretical analysis under the SS-ANOVA framework.

For the multivariate η ($d > 1$), multiple smoothing parameters are involved in adjusting the "strength" of the corresponding components in the functional ANOVA decomposition. Gu and Wahba (1991) proposed to select multiple smoothing parameters by minimizing GCV/GML through a modified Newton method. With all smoothing parameters tunable, the iterative algorithm takes $O(Sn^3)$ flops per iteration, where S is the number of smoothing

parameters, and needs tens of iterations to converge. The algorithm is quite efficient when S is small. However, as the number of multi-way interaction components increases, the number of smoothing parameters grows dramatically. For instance, S equals 5 for the full two-way model and S equals 19 for the full three-way model. Thus, the algorithm is computationally expensive for multi-dimensional models with interactions. Several methods were proposed to ameliorate the heavy computing burden in the model with interactions. The obvious option is providing good pre-specified values for multiple smoothing parameters. Gu and Wahba (1991) proposed an algorithm to calculate these values and showed the minimizer of (1.2) based on them usually yielded good estimates. Although the algorithm performs well in additive models, it is unreliable when there are interactions. Such an unreliable performance may be aggravated when the model is misspecified. Other than removing the iterations, Helwig and Ma (2015) proposed a reparameterization of smoothing parameters in SS-ANOVA models. For the reparameterization, there is one smoothing parameter for each predictor and the smoothing parameter for the interaction term is the product of smoothing parameters of the corresponding predictors. Based on the reparameterization, the new algorithm has computational costs comparable to those of generalized additive models (GAM) (Hastie and Tibshirani, 1986). However, the algorithm may produce a biased estimate when the SS-ANOVA model is misspecified. In addition, its theoretical foundation calls for further justification. Parallel to the work under the framework of SS-ANOVA, Wood (2000) proposed an efficient smoothing parameters selection method for GAM. A more stable version was proposed to deal with the rank deficiency of the GAM fitting problem (Wood, 2004). Although these procedures have been applied to GAM successfully, they have several limitations due to well-known drawbacks of GAM. One of the most well-known ones is the difficulty to include interactions in GAM, and therefore, these smoothing parameters selection methods can not be used when including the interactions is necessary. The situation is abundant in genomic applications (Sun et al., 2016).

Except for the methods from the computational perspective, the asymptotic behaviors

of $\eta_{n,\lambda}$ and the optimal λ have been studied by a number of authors, see Silverman (1982), Rice and Rosenblatt (1983), Cox (1984), Speckman (1985), Cox and O'Sullivan (1990), and Gu and Qiu (1993). The estimator can achieve an optimal convergence rate when the smoothing parameter is of order $O(n^{-r/(pr+1)})$ for $r > 1$ and $p \in [1, 2]$. Lin (2000) further studied the optimal convergence rate of the estimator in tensor product space ANOVA models and showed the optimal rate of smoothing parameters depended on the highest order of interactions. In real applications, one may directly use $Cn^{-r/(pr+1)}$ for some pre-defined C, r, and p as the smoothing parameter when the sample size is n (Hall, 1990). However, its numerical performance is unreliable, which is observed in our simulation study.

Despite the abundance of research from computational and theoretical perspectives individually, few studies combine these two perspectives together. One of the exceptions in selecting the threshold for wavelet shrinkage estimators is reviewed below. The threshold is similar to the smoothing parameter in SS-ANOVA models, and improper choice of the threshold may let the shrinkage estimators overfit or underfit the data. Nason (1996) proposed a modified twofold cross-validation algorithm to select the threshold. In the modified algorithm, the threshold for the full samples of size n is estimated by multiplying the threshold for the subset of size $n/2$ with a theoretically justified correction term related to n. The algorithm works well in in practice. However, it is computationally infeasible to choose the thresholds for high-dimensional cases since the computational complexities depend exponentially on d.

To make the smoothing parameters selection practical in large samples, I develop an *asympirical* (*asym*ptotic + em*pirical*) smoothing parameters selection method taking advantages of theoretical properties of smoothing parameters and aforementioned computational methods for SS-ANOVA models. In the proposed method, I choose a subsample of size to be much less than the full sample size n, and select smoothing parameters for the subsample using the GCV method. The smoothing parameters for the full sample are extrapolated based on the selected smoothing parameters and the optimal rate $O(n^{-r/(pr+1)})$. The pro-

posed smoothing parameters selection method reduces the computational complexity from tens of $O(Sn^3)$ flops, which is required by GCV/GML, to $O(B^3)$, where B is the size of subsamples. The numerical advantage of the proposed algorithm over the other approaches is much more significant when there are multiple interactions (large S) in the model. Besides the numerical advantages, the proposed smoothing parameters share optimal properties with the ones minimizing a risk function for full samples. Furthermore, the estimator, η_{n,λ^*}, based on the proposed smoothing parameters λ^* attains the optimal convergence rate.

1.2 Overview of Complex Data Issues

Optimal Penalized Function-on-Function Regression

For the complex datasets collected in many time-course studies, most of the existing literature has only considered the regression models of a scalar response against one or more functional predictors, possibly with some scalar predictors as well. Some of them considered a reproducing kernel Hilbert space (RKHS) framework. For example, Yuan and Cai (2010) provided a thorough theoretical analysis of the penalized functional linear regression model with a scalar response. The paper laid the foundation for several theoretical developments including the Representer Theorem and minimax convergence rates for prediction and estimation for penalized functional linear regression models. In a follow-up, Cai and Yuan (2012) showed that the minimax rate of convergence for the excess prediction risk is determined by both the covariance kernel and the reproducing kernel. Then they designed a data-driven roughness regularization predictor that can achieve the optimal convergence rate adaptively without the knowledge of the covariance kernel. Du and Wang (2014) extended the work of Yuan and Cai (2010) to the setting of a generalized functional linear model, where the scalar response comes from an exponential family distribution.

In contrast to these functional linear regression models with a scalar response, the model with a functional response $Y(t)$ over a functional predictor $X(s)$ has only been scarcely

investigated (Yao et al., 2005b; Ramsay and Silverman, 2005). Such data with functional responses and predictors are abundant in practice. I shall now present two motivating examples.

Example 1.2.1 ***Canadian Weather Data***

Daily temperature and precipitation at 35 different locations in Canada averaged over 1960 to 1994 were collected (Figure 1.1). The main interest is to use the daily temperature profile to predict the daily precipitation profile for a location in Canada.

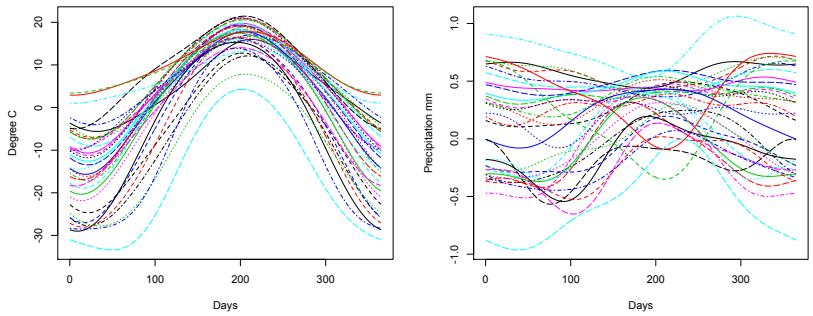

Figure 1.1: Smoothed trajectories of temperature (Celsius) in left panel and the log (base 10) of daily precipitation (Millimeter) in right panel. The x-axis labels in both panels represent 365 days.

Example 1.2.2 ***Histone Regulation Data***

Extensive researches have been shown that histone variants, i.e. histones with structural changes compared to their primary sequence, play an important role in the regulation of chromatin metabolism and gene activity (Ausió, 2006). An ultra-high throughput time course experiment was conducted to study the regulation mechanism during heat stress in Arabidopsis thaliana. The genome-wide histone variant distribution was measured by ChIP sequencing (ChIP-seq) (Johnson et al., 2007) experiments. We computed histone levels over 350 base pairs (bp) on genomes from the ChIP-seq data, see left panel in Figure 1.2. The RNA sequencing (RNA-seq) (Wang et al., 2009) experiments measured the expression levels over seven time points within 24 hours, see right panel in Figure 1.2. Of primary interest is to study the regulation mechanism between gene expression levels over time domain and histone

levels over spatial domain.

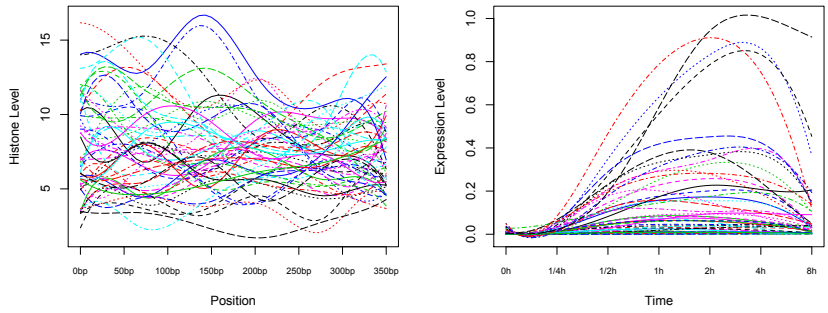

Figure 1.2: Smoothed trajectories of normalized histone levels in ChIP-seq experiments in left panel and the normalized expression levels in RNA-seq experiments in right panel. The x-axis label in the left panel stands for the region of 350 bp. The x-axis label in the right panel represents seven time points within 24 hours.

Motivated by the examples, we now present the statistical model. Let $\{(X(s), Y(t)) : s \in I_x, t \in I_y$ be two random processes defined respectively on $I_x, I_y \subseteq \mathbb{R}$. Suppose n independent copies of $\big(X, Y\big)$ are observed: $\big(X_i(s), Y_i(t)\big)$, $i = 1, \ldots, n$. The functional linear regression model of interest is

$$Y_i(t) = \alpha(t) + \int_{I_x} \beta\big(t, s\big) X_i(s) ds + \epsilon_i(t), \qquad t \in I_y, \tag{1.3}$$

where $\alpha(\cdot) : I_y \to \mathbb{R}$ is the intercept function, $\beta(\cdot,\ \cdot) : I_y \times I_x \to \mathbb{R}$ is a bivariate coefficient function, and $\epsilon_i(t)$, independent of $X_i(s)$, are i.i.d. random error functions with $\mathbb{E}\epsilon_i(t) = 0$ and $\mathbb{E}\|\epsilon_i(t)\|_2^2 < \infty$. In Example 1.2.1, $Y_i(t)$ and $X_i(t)$ represent the daily precipitation and temperature at station i. In Example 1.2.2, the expression levels of gene i over seven time points, $Y_i(t)$, from RNA-seq is used as the functional response. The histone levels of gene i over 350 base pairs (bp), $X_i(s)$, from ChIP-seq is used as the functional predictor.

At a first look, model (1.3) might give the (wrong) impression of being an easy extension from the model with a scalar response, with the latter obtained from (1.3) by removing all the t notation. However, the coefficient function in the scalar response case is univariate and thus can be easily estimated by most off-the-shelf smoothing methods. When extended to

estimating a bivariate coefficient function $\beta(t, s)$ in (1.3), many of these smoothing methods may encounter major numerical and/or theoretical difficulties. This partly explains the much less abundance of research in this direction.

Some exceptions though are reviewed below. Cuevas et al. (2002) considered a fixed design case, a different setting from (1.3) with $Y_i(t)$ and $X_i(s)$ represented and analyzed as sequences. Nonetheless they provided many motivating applications in neuroscience, signal transmission, pharmacology, and chemometrics, where (1.3) can apply. The historical functional linear model in Malfait and Ramsay (2003) was among the first to study regression of a response functional variable over a predictor functional variable, or more precisely, the history of the predictor function. Ferraty et al. (2011) proposed a simple extension of the classical Nadaraya-Watson estimator to the functional case and derived its convergence rates. They provided no numerical results on the empirical performance of their kernel estimator. Benatia et al. (2017) extended ridge regression to the functional setting. However, their estimation relied on an empirical estimate of the covariance process of predictor functions. Theoretically sound as it is, this covariance process estimate is generally not reliable in practice. Consequently, their coefficient surface estimates suffered as shown in their simulation plots. Meyer et al. (2015) proposed a Bayesian function-on-function regression model for multi-level functional data, where the basis expansions of functional parameters were regularized by basis-space prior distributions and a random effect function was introduced to incorporate the with-subject correlation between functional observations.

A popular approach has been the functional principal component analysis (FPCA) as in Yao et al. (2005b) and Crambes and Mas (2013). The approach starts with a basis representation of $\beta(t, s)$ in terms of the eigenfunctions in the Karhunen-Loève expansions of $Y(t)$ and $X(s)$. Since this representation has infinitely many terms, it is truncated at certain point to obtain an estimable basis expansion of $\beta(t, s)$. Yao et al. (2005b) studied a general data setting where $Y(t)$ and $X(s)$ are only sparsely observed at some random points. They derived the consistency and proposed asymptotic pointwise confidence bands for predicting

response trajectories. Crambes and Mas (2013) furthered the theoretical investigation of the FPCA approach by providing a minimax optimal rates in terms of the mean square prediction error. However, the FPCA approach has a couple of critical drawbacks. Firstly, $\beta(t, s)$ is a statistical quantity unrelated to $Y(t)$ or $X(s)$. Hence the leading eigenfunctions in the truncated Karhunen-Loève expansions of $Y(t)$ and $X(s)$ may not be an effective basis for representing $\beta(t, s)$. See, e.g., Cai and Yuan (2012) and Du and Wang (2014) for some scalar-response examples where the FPCA approach breaks down when the aforementioned situation happens. Secondly, the truncation point is integer-valued and thus only has a discrete control on the model complexity. This puts it at disadvantage against the roughness penalty regularization approach, which offers a continuous control via a positive and real-valued smoothing parameter (Ramsay and Silverman, 2005, Chapter 5).

In this thesis, I consider a penalized function-on-function regression approach to estimating the bivariate coefficient function $\beta(t, s)$. There have been a few recent developments in the direction of penalized function-on-function regression. Lian (2015) studied the convergence rates of the function-on-function regression model under a RKHS framework. Although his model resembled model (1.3), he developed everything with the variable t fixed and did not enforce any regularization on the t direction. Firstly, this lack of t-regularization can be problematic since this leaves the noisy errors on the t direction completely uncontrolled and can result in an $\beta(s, t)$ estimate that is very rough on the t direction. Secondly, this simplification of fixing t essentially reduces the problem to a functional linear model with a scalar response and thus makes all the results in Yuan and Cai (2010) directly transferrable even without calling on any new proofs. The R package `fda` maintained by Ramsay et al. has implemented a version of penalized B-spline estimation of $\beta(t, s)$ with a fixed smoothing parameter. Ivanescu et al. (2015) considered a penalized function-on-function regression model where the coefficient functions were represented by expansions into some basis system such as tensor cubic B-splines. Quadratic penalties on the expansion coefficients were used to control the smoothness of the estimates. This work provided a nice multiple-predictor-

function extension to the function-on-function regression model in the `fda` package. Scheipl and Greven (2016) studied the identifiability issue in these penalized function-on-function regression models. However, this penalized B-spline approach has several well-known drawbacks. First, it is difficult to show any theoretical optimality such as the minimax risk of mean prediction in Cai and Yuan (2012). So its theoretical soundness is hard to justify. Moreover, the B-spline expansion is only an approximate solution to the optimization of the penalized least squares score. Hence the penalized B-spline estimate is not numerically optimal from the beginning either. These drawbacks can have negative impacts on the numerical performance as we shall see from the simulation results.

The penalized function-on-function regression method proposed in this thesis obtains its estimator of $\beta(t, s)$ through the minimization of penalized least squares on a RKHS that is naturally associated with the roughness penalty. Such a natural formulation through a RKHS offers several advantages comparing with the existing penalized function-on-function regression methods. Firstly, it allows us to establish a Representer Theorem which states that, although the optimization of the penalized least squares is defined on an infinite dimensional function space, its solution actually resides in a data-adaptive finite dimensional subspace. This result guarantees an exact solution when the optimization is carried out on this finite dimensional subspace. This result itself is a nontrivial generalization of the Representer Theorems in the scenarios of nonparametric smooth regression model (Wahba, 1990) and the penalized functional regression model with a scalar response (Yuan and Cai, 2010). Based on the Representer Theorem, I propose an estimation algorithm which uses penalized least squares and Gaussian quadrature with the Gauss-Legendre rule to estimate the bivariate coefficient function. The smoothing parameter is selected by the generalized cross validation (GCV) method. Secondly, the RKHS framework allows us to show that our estimator has the optimal rate of mean prediction since it achieves the minimax convergence rate in terms of the excess risk. This generalizes the results in Cai and Yuan (2012) and Du and Wang (2014) for functional linear regression with a scalar response to the functional

response scenario. In the numerical study, I have also considered the problem with sparsely sampled data. Particularly, I introduce an extra pre-smoothing step before applying the proposed penalized functional regression model. The pre-smoothing step implements the principal-component-analysis-through-expectation (PACE) method in Yao et al. (2005a). The extensive simulation studies demonstrate the numerical advantages of the proposed method over the existing ones. In summary, the proposed method has the following distinguishing features: (i) it makes no structural dependence assumptions of $\beta(t, s)$ over the predictor and response processes; (ii) the Representer Theorem guarantees an exact solution instead of an approximation to the optimization of the penalized score; (iii) benefited from the Representer Theorem, I develop a numerically reliable algorithm that has sound performance in simulations; (iv) the estimator achieves the optimal minimax convergence rate in mean prediction.

Identification of Differentially Expressed Genes

Accurately identifying differentially expressed (DE) genes in time course RNA-seq data is crucial for understanding the dynamics of transcriptional regulatory network. Inferring DE genes in time course RNA-seq experiments has a number of interesting challenges. First, the DE genes in time course data are those with different gene expression profiles along the time across treatments or conditions. However, most of the available methods treat expressions of a gene at different time points as replicates and test the significance of the mean expression difference between treatments or conditions irrespective of time, e.g., edgeR (Robinson et al., 2010) and DESeq (Anders and Huber, 2010). They thus fail to identify many DE genes with different profiles across time. Second, some methods have been developed recently to identify the DE genes with different expression profiles over time. A recent work by Oh et al. (2013) models time dependency using a hidden Markov model. Such a model requires the Markov property. In particular, the Markov property states that the conditional dependency of prior information from all time can be simplified to the conditional dependency of prior information

of k time points (kth order Markov chain). It is still unclear whether such Markov property holds for general time course RNA-seq data. Finally, both edgeR and DESeq use the total read counts of each gene and model the variation of the read counts across the replicates at gene level. When RNA-seq experiments do not have replicates or the number of replicates is small, the statistical significance tests in edgeR and DESeq have small degrees of freedom and may result in a high false discovery rate (FDR).

To surmount these challenges, I develop a novel nonparametric method to identify DE genes in this thesis. The input of the proposed method is the read counts at the exon level for each gene at each time point. The read counts of genes at the exon level across different time points are modeled by a negative binomial mixed-effect model (NBMM). In this model, the mean gene expression profiles over time across treatments are modeled by a nonparametric bivariate function of time and treatments, while the time dependency is characterized by a parametric random effect. The nonparametric bivariate function has

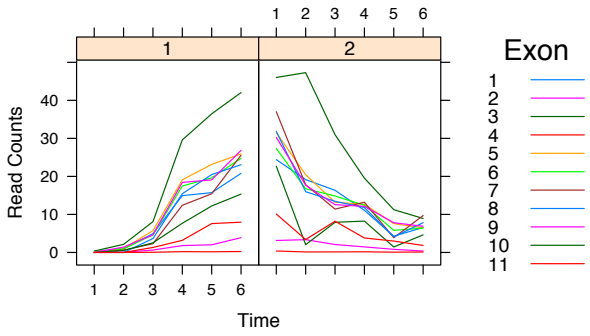

Figure 1.3: Gene *ss* (FlyBase ID: FBgn0003513) was identified as non-parallel differentially expressed with p value=0.00. Different exons are represented by curves with varying colors. This gene participates in antennal development, antennal morphogenesis, and imaginal disc-derived leg segmentation. Read counts on the y-axis are the average counts (The total read counts on each exon divided by the length of exon). The left panel and right panel represent the early and late embryonic developmental stages respectively.

great flexibility in modeling different expression profiles over possibly non-equally spaced time points across treatments and conditions. The parametric random effects are used to define a variety of time dependency correlation structures. The model is fitted by a penalized

likelihood method. In order to identify DE genes unique to time course experiments, we define two types of DE genes in time course RNA-seq experiments: nonparallel differentially expressed (NPDE) genes with nonparallel expression profiles over time across treatments, see Figure 1.3, and parallel differentially expressed (PDE) genes with parallel expression profiles over time across treatments, see Figure 1.4. PDE genes are those consistently up-

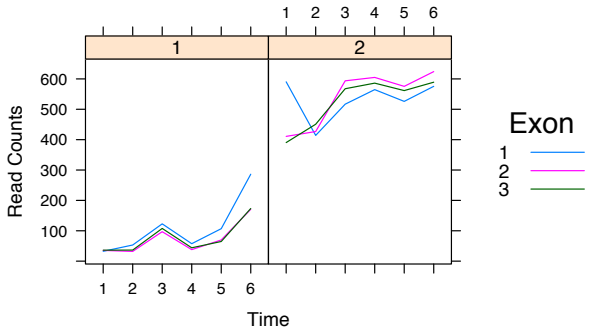

Figure 1.4: Gene $Idgf2$ (FlyBase ID: FBgn0020415) was identified as parallel differentially expressed with p value=0.00. Different exons are represented by curves with varying colors. This gene participates in imaginal disc development. Read counts on the y-axis are the average counts (The total read counts on each exon divided by the length of exon). The left panel and right panel represent the early and late embryonic developmental stages respectively.

regulated or down-regulated over time across treatments, whereas NPDE genes are those that have significant expression profile changes over time across treatments. Compared with PDE genes, in many scientific investigations, NPDE genes are of primary interest. Focused study of the NPDE genes may provide more information on how the cell responds differently to different stimulus or treatments. Moreover, time course RNA-seq experiments are commonly used in case-control studies and in clinical trials. In such experiments, mRNA samples are taken from a small number of subjects over time in the treatment group and from another small number of subjects in the control group. Because each group only consists of a small number of subjects, one subject with high baseline gene expression can cause a high average gene expression for the whole group. Thus, there are many PDE genes between treatments, but they are biologically irrelevant (Ma et al., 2009). To distinguish the two types of DE genes, I decompose the nonparametric bivariate function in the model into

the main effects of time and treatment separately, as well as their interaction through a functional ANOVA decomposition. The identification of DE genes is equivalent to testing significance of treatment-time interactions in the functional ANOVA decomposition. I fit this model to the exon level read counts data using penalized maximum likelihood. The tuning parameter is selected by cross-validation (Gu and Ma, 2005).

Chapter 2

Prerequisite Definitions

Chapter Summary: In this chapter, I define some fundamental concepts used throughout the thesis. In particular, the definitions of linear space, inner products, and reproducing kernel Hilbert space (RKHS) are discussed. In addition, the functional ANOVA decomposition in a Hilbert space, which is crucial for the proposed methods, is reviewed. For more information on the topics, see Gu (2013); Wahba (1990).

2.1 Linear Spaces

For elements f, g, h, define the operation of addition satisfying the following three properties:

1. $f + g = g + f$,
2. $(f + g) + h = f + (g + h)$,
3. $\forall f, g$, there exists an element such that $f + h = g$.

Similarly, we define the operation of scalar multiplication satisfying the following four properties:

1. $\alpha(f + g) = \alpha f + \alpha g$,
2. $(\alpha + \beta)f = \alpha f + \beta f$,
3. $1f = f$,
4. $0f = 0$,

where $\alpha, \beta \in \mathbb{R}$. A set $\mathcal{L}$ of such elements forms a linear space if $f, g \in \mathcal{L}$ satisfying the following two conditions:

1. $f + g \in \mathcal{L}$,

2. $\alpha f \in \mathcal{L}$,

for any scalars $\alpha \in \mathbb{R}$. A functional L in a linear space operates on an element $f \in \mathcal{L}$ and produces a real number as its value ($L : \mathcal{L} \to \mathbb{R}$). A linear functional in $\mathcal{L}$ is a functional satisfying the following two conditions:

1. $L(f + g) = Lf + Lg$,

2. $L(\alpha f) = \alpha L f$,

for $f, g \in \mathcal{L}$ and $\alpha \in \mathbb{R}$. Analogously, a bilinear form J takes in two elements as arguments and returns a real number, i.e., $J(f, g) = \alpha$ for some $\alpha \in \mathbb{R}$. A bilinear form satisfies the following two properties:

1. $J(\alpha f + \beta g, h) = \alpha J(f, h) + \beta J(g, h)$,

2. $J(f, \alpha g + \beta h) = \alpha J(f, g) + \beta J(f, h)$,

for $f, g, h \in \mathcal{L}$ and $\alpha, \beta \in \mathbb{R}$. Based on the above properties, the bilinear form reduces to a linear functional in the other argument if one argument is fixed. If $J(f, g) = J(g, f)$ for $f, g \in \mathcal{L}$, the $J(\cdot, \cdot)$ is said to be symmetric. Furthermore, a symmetric bilinear form is said to be nonnegative definite if $J(f, f) \geq 0$, $\forall f \in \mathcal{L}$, and positive definite if $J(f, f) > 0$, $\forall f \in \mathcal{L}$ ($f \neq 0$). A quadratic functional is nonnegative definite and often denoted as $J(f)$.

A linear space is typically equipped with inner products, which is a positive definite bilinear form with a notation $\langle \cdot, \cdot \rangle$. An inner product defines a norm, $\|f\| = \sqrt{\langle f, f \rangle}$, in the linear space. This definition provides a metric to measure the distance in the space, defined as $D(f, g) = \|f - g\|$. In such a linear space, two inequalities are quite useful (Cauchy-Schwarz and triangle inequalities):

1. $|\langle f, g\rangle| \leq \|f\| \times \|g\|$,

2. $\|f + g\| \leq \|f\| + \|g\|$,

where the equality will hold if and only if $f = \alpha g$ for some $\alpha \in \mathbb{R}$ (Cauchy-Schwarz inequality) and for $\alpha > 0$ (triangle inequality). If $\lim_{n\to\infty} \|f_n - f\| = 0$ for a sequence of elements f_n, the sequence is said to converge to its limit points f, denoted by $\lim_{n\to\infty} f_n = f$. If $\lim_{n\to\infty} Lf_n = Lf$ whenever $\lim_{n\to\infty} f_n = f$, the functional L is said to continuous. A Cauchy sequence is a sequence that the elements in it become closer and closer as $n \to \infty$, i.e., $\lim_{m,n\to\infty} \|f_m - f_n\| = 0$. If every Cauchy sequence in a linear space $\mathcal{L}$ converges to an element in $\mathcal{L}$, the space is said to be complete. An element is a limit point of a set if it is the limit point of sequence in the set. This set is closed if the set contains all of its own limit points.

2.2 Reproducing Kernel Hilbert Spaces

If a linear space is complete and equipped with an inner product, the space is said to be a Hilbert Space, and can be thought of as a generalization of the familiar Euclidean space. In a general Hilbert space denoted by $\mathcal{H}$, the continuity of a functional is not always satisfied. To solve the problem, we need a special Hilbert space, i.e., RKHS.

For each $g \in \mathcal{H}$, there exists a corresponding continuous linear functional L_g such that $L_g(f) = \langle g, f\rangle$. Conversely, an element $g_L \in \mathcal{H}$ can also be found such that $\langle g_L, f\rangle = L(f)$ for any continuous linear functional L in $\mathcal{H}$. This is known as the Riesz representation theorem.

Theorem 2.2.1 (Riesz Representation Theorem) *Let $\mathcal{H}$ be a Hilbert space. For any functional L of $\mathcal{H}$, there uniquely exists an element $g_L \in \mathcal{H}$ such that*

$$L(\cdot) = \langle g_L, \cdot\rangle,$$

where g_L is called the representer of L. The uniqueness is in the sense that g_1 and g_2 are

considered as the same representer for any g_1 and g_2 satisfying $||g_1 - g_2|| = 0$.

We then have the following definition of RKHS.

Definition 1 (Reproducing Kernel Hilbert Space) *Consider a Hilbert space $\mathcal{H}$ consisting of real-valued functions f on the domain $\mathcal{X}$. For every element $x \in \mathcal{X}$, define an evaluation functional L_x such that $L_x(f) = f(x)$. If all the evaluation functional are continuous, $\forall x \in \mathcal{X}$, then $\mathcal{H}$ is referred to as RKHS.*

Roughly speaking, the continuity of evaluation functional means that if two functions f and g are close in norm, i.e., $\|f - g\|$ is small, then f and g are also pointwise close, i.e., $|f(x) - g(x)|$ is small for all x.

By Theorem 2.2.1, for every evaluation functional L_x, there exists a corresponding function $R_x \in \mathcal{H}$ on $\mathcal{X}$ as the representer of evaluation functional, such that $\langle R_x, f\rangle = f(x)$, $\forall f \in \mathcal{H}$. By the definition of evaluation functional, it follows

$$R_x(y) = \langle R_x, R_y\rangle = R_y(x).$$

The bivariate function $R(x, y) = \langle R_x, R_y\rangle$ is called the reproducing kernel of $\mathcal{H}$, which is unique if it exists. The essential meaning of the name "reproducing kernel" comes from its reproducing property

$$\langle R_x(\cdot), f\rangle = f(x)$$

for any $f \in \mathcal{H}$. In general, a RKHS defines a reproducing kernel function that is both symmetric and positive definite. In addition, Moore-Aronszajn theorem states that every symmetric, positive definite kernel defines a unique RKHS (Aronszajn, 1950), and hence one can construct a RKHS simply by specifying its reproducing kernel.

To construct the RKHS on a product domain $\prod_{j=1}^{d} \mathcal{X}_j$, one may take the tensor product of spaces for the marginal domains $\mathcal{X}_j$. This construction relies on the following theorem.

Theorem 2.2.2 *For non-negative definite $R_{\mathcal{X}_1}$ and $R_{\mathcal{X}_2}$ on $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively, $R_{\mathcal{X}} = R_{\mathcal{X}_1} R_{\mathcal{X}_2}$ is non-negative definite on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$.*

Suppose that $\mathcal{H}_{\mathcal{X}_1}$ and $\mathcal{H}_{\mathcal{X}_2}$ are RKHS of functions with reproducing kernels $R_{\mathcal{X}_1}$ and $R_{\mathcal{X}_2}$. Given that $R_{\mathcal{X}} = R_{\mathcal{X}_1} R_{\mathcal{X}_2}$ is non-negative definite on $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$, the function $R_{\mathcal{X}}$ is the reproducing kernel for a RKHS on $\mathcal{X}$. The RKHS corresponding to $R_{\mathcal{X}}$ is known as the tensor product RKHS denoted by $\mathcal{H} = \mathcal{H}_{\mathcal{X}_1} \otimes \mathcal{H}_{\mathcal{X}_2}$. This operation can be applied recursively to create a tensor product RKHS on the product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$, such as $\mathcal{H} = \otimes_{j=1}^{d} \mathcal{H}_{\mathcal{X}_j}$.

2.3 ANOVA Decompositions

One-way ANOVA Decomposition

We consider a classical one-way ANOVA model $y_{ij} = \mu_i + \epsilon_{ij}$, where y_{ij} is the observed data, μ_i is the treatment mean for $i = 1, ..., I$, $j = 1, ..., J$, ϵ_{ij}'s are the random errors. The treatment mean μ_i can be further decomposed as $\mu_i = \mu + \alpha_i$, where μ is the overall mean and α_i is the treatment effect with the constraint $\sum_{i=1}^{I} \alpha_i = 0$. Similar to the classical ANOVA decomposition, a univariate function f can be decomposed as

$$f = Af + (I - A)f = f_c + f_x, \tag{2.1}$$

where A is an averaging operator that averages the effect of x, I is an identity operator. The operator A averages a function f to a constant function f_c satisfying $A(I - A) = 0$. In (2.1), $f_c = Af$ is the mean function, and $f_x = (I - A)f$ is the treatment effect.

Multi-way ANOVA Decomposition

On a d-dimensional product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$, a multivariate function $f(x_{\langle 1 \rangle}, ..., x_{\langle d \rangle})$ can be decomposed similarly to the one-way ANOVA decomposition. Let A_j, $j = 1, ..., d$,

be the average operator on $\mathcal{X}_j$, then $A_j f$ is a constant function on $\mathcal{X}_j$. One can define the ANOVA decomposition on $\mathcal{X}$ as

$$\begin{aligned} f &= \{\prod_{j=1}^{d}(I - A_j + A_j)\}f \\ &= \sum_{\mathcal{S}}\{\prod_{j\in\mathcal{S}}(I - A_j)\prod_{j\notin\mathcal{S}}A_j\}f = \sum_{\mathcal{S}} f_{\mathcal{S}}, \end{aligned} \tag{2.2}$$

where $\mathcal{S} \subseteq \{1, ..., d\}$. The term $f_c = \prod_{j=1}^{d} A_j f$ is the constant function, $f_j = (I - A_j)\prod_{\alpha\neq j} A_\alpha f$ is the main effect term of $x_{\langle j\rangle}$, and the term $f_{\mu\nu} = (I - A_\mu)(I - A_\nu)\prod_{\alpha\neq\mu,\nu} A_\alpha f$ is the interaction of $x_{\langle\mu\rangle}$ and $x_{\langle\nu\rangle}$, and so on.

Chapter 3

Asympirical Smoothing Parameters Selection in Big Data

Chapter Summary: In this chapter, I develop an *asympirical* (*asymp*totic + em*pirical*) smoothing parameters selection approach for smoothing spline ANOVA models in big data. In the approach, I perform the asymptotic analysis to show optimal smoothing parameters are the polynomial function of the sample size and an unknown constant. The unknown constant is then estimated through the empirical subsample extrapolation. The proposed method can significantly reduce computational costs of selecting smoothing parameters in high-dimensional and large samples. I show smoothing parameters chosen by the proposed method tend to the optimal smoothing parameters minimizing a risk function. In addition, the estimator based on the proposed smoothing parameters achieves the optimal convergence rate. Extensive simulation studies demonstrate numerical advantages of the proposed method over competing methods in the selection of smoothing parameters. The proposed method is then applied to two real data examples.

3.1 Smoothing Spline ANOVA Models

In this section, I review the Kimeldorf-Wahba Representer Theorem (Kimeldorf and Wahba, 1971; Wahba, 1990; Wang, 2011) which ensures that the solution of the penalized least squares defined in the infinite dimensional functional space actually resides in a finite dimensional space. Then, several roughness penalties used in the estimation are presented. In the end, the method for multiple smoothing parameters selection is reviewed.

Estimation

Recall that the minimization of (1.2) is performed in the tensor product RKHS $\mathcal{H} = \{\eta : J(\eta) < \infty\}$. The quadratic roughness penalty $J(\eta) = \sum_{\delta=1}^{S} \theta_\delta^{-1} \langle \eta, \eta \rangle_\delta$, where θ_δ's are smoothing parameters adjusting the "strength" of the corresponding components, $\langle \cdot, \cdot \rangle_\delta$ is the inner product in $\mathcal{H}_\delta$ with reproducing kernel $R_\delta(\cdot, \cdot)$, and S is the number of subspaces based on the ANOVA decomposition, see details in Chapter 2. The space $\mathcal{H}$ has the tensor sum decomposition $\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J$, where the null space of $\mathcal{H}$, $\mathcal{N}_J$, is spanned by $\{\phi_\nu\}_{\nu=1}^{M}$ and $R(\cdot, \cdot) = \sum_{\delta=1}^{S} \theta_\delta R_\delta(\cdot, \cdot)$ is the reproducing kernel of $\mathcal{H}_J = \oplus_{\delta=1}^{S} \mathcal{H}_\delta$.

Theorem 3.1.1 *(Kimeldorf-Wahba Representer Theorem) The minimizer of* (1.2) *is given by*

$$\eta(x) = \sum_{\nu=1}^{M} d_\nu \phi_\nu(x) + \sum_{i=1}^{n} c_i R(x_i, x),$$

where $\mathbf{d} = (d_1, \cdots, d_M)^t$, *and* $\mathbf{c} = (c_1, \cdots, c_n)^t$ *are unknown coefficients.*

Theorem 3.1.1 facilitates the estimation by reducing an infinite dimensional optimization problem to a finite dimensional one. Based on the Representer Theorem, the minimization in (1.2) becomes

$$(\mathbf{Y} - T\mathbf{d} - K\mathbf{c})^t (\mathbf{Y} - T\mathbf{d} - K\mathbf{c}) + n\lambda \mathbf{c}^t K \mathbf{c}, \tag{3.1}$$

where $\mathbf{Y} = (y_1, \cdots, y_n)^t$, $T_{n\times M}$ is a matrix with (i,ν)th entry $\phi_\nu(x_i)$, and $K_{n\times n}$ is a matrix with (i,j)th entry $R(x_i, x_j)$. Differentiating (3.1) with respect to $\mathbf{d}$ and $\mathbf{c}$ and setting the derivatives to zero, one obtains the following linear system of equations

$$\begin{pmatrix} T^tT & T^tK \\ K^tT & K^tK + n\lambda K \end{pmatrix} \begin{pmatrix} \mathbf{d} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} T^t\mathbf{Y} \\ K^t\mathbf{Y} \end{pmatrix}. \tag{3.2}$$

To estimate $\mathbf{d}$ and $\mathbf{c}$, one needs to solve the linear system (3.2). If smoothing parameters λ, θ_δ's are known, the computational cost is typically $O(n^3)$.

Roughness Penalties

One can choose different forms of roughness penalties. The most popular one for the univariate η on $\mathcal{X}$ is

$$J(\eta) = \int_{\mathcal{X}} (\eta^{(m)})^2 dx,$$

where $\eta^{(m)} = d^m\eta/dx^m$. Setting $m = 2$, a cubic spline estimator is obtained by minimizing (1.2) (Wahba, 1990).

When estimating multivariate functions on $\mathcal{X} \subset \mathcal{R}^d$, one can use the thin-plate spline penalty

$$J_m^d(\eta) = \sum_{\tau_1+\cdots+\tau_d=m} \frac{m!}{\tau_1!\cdots\tau_d!} \times \int \cdots \int_{\mathcal{X}} \left(\frac{\partial^m \eta}{\partial x_{\langle 1\rangle}^{\tau_1} \cdots \partial x_{\langle d\rangle}^{\tau_d}} \right)^2 dx_{\langle 1\rangle} \cdots dx_{\langle d\rangle},$$

where m is the order of derivatives (Duchon, 1977). In particular, the cubic thin-plate spline penalty for two-dimensional spatial data has the form,

$$J_2^2(\eta) = \int \int_{\mathcal{X}} \left(\frac{\partial^2 \eta}{\partial x_{\langle 1\rangle}^2} \right)^2 + 2\left(\frac{\partial^2 \eta}{\partial x_{\langle 1\rangle} x_{\langle 2\rangle}} \right)^2 + \left(\frac{\partial^2 \eta}{\partial x_{\langle 2\rangle}^2} \right)^2 dx_{\langle 1\rangle} dx_{\langle 2\rangle},$$

where $x_{\langle 1\rangle}$ and $x_{\langle 2\rangle}$ represent, for instance, longitude and latitude coordinates. Since the thin-plate spline is invariant to the rotation of coordinates, it is a popular tool to model

spatial data.

For multivariate functions η, we can decompose it into

$$\eta(x) = \eta_{\emptyset} + \sum_{j=1}^{d} \eta_j(x_{\langle j \rangle}) + \sum_{j<k} \eta_{jk}(x_{\langle j \rangle}, x_{\langle k \rangle}) + \cdots + \eta_{12\cdots d}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}, \cdots, x_{\langle d \rangle}), \tag{3.3}$$

where $\eta_{\emptyset}$ is a constant, η_j's are the main effects, η_{jk}'s are the two-way interactions, and $\eta_{12\cdots d}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle}, \cdots, x_{\langle d \rangle})$ is the d-way interaction. Side conditions are imposed to the components to guarantee the uniqueness. Another convenient way to define the penalty for the multivariate function is to construct the tensor product RKHS. The RKHS $\mathcal{H}$ can be decomposed into the space of constant, spaces of main effects, and the corresponding spaces of interaction terms, see (3.3), lying in the tensor product space of the interacting main-effect spaces.

Example 3.1.1 *For the tensor product cubic spline on* $[0,1]^2$*, one has the space decomposition on each domain*

$$\begin{aligned} \{f : f^{(2)} \in L_2[0,1]\} &= \{f : f \propto 1\} \oplus \{f : f \propto k_1\} \\ \oplus \{f : \int_0^1 f dx = \int_0^1 f^{(1)} dx = 0, f^{(2)} &\in L_2[0,1]\} \\ &= \mathcal{H}_{00} \oplus \mathcal{H}_{01} \oplus \mathcal{H}_1, \end{aligned}$$

where $k_1(x) = x - 0.5$*. The space of constant term is* $\mathcal{H}_{00\langle 1 \rangle} \otimes \mathcal{H}_{00\langle 2 \rangle}$*, and the* $\mathcal{H}_{00\langle 1 \rangle} \otimes (\mathcal{H}_{01\langle 2 \rangle} \oplus \mathcal{H}_{1\langle 2 \rangle})$ *and* $\mathcal{H}_{00\langle 2 \rangle} \otimes (\mathcal{H}_{01\langle 1 \rangle} \oplus \mathcal{H}_{1\langle 1 \rangle})$ *span the space of main effects, and the subspace* $(\mathcal{H}_{01\langle 1 \rangle} \oplus \mathcal{H}_{1\langle 1 \rangle}) \otimes (\mathcal{H}_{01\langle 2 \rangle} \oplus \mathcal{H}_{1\langle 2 \rangle})$ *spans the space of interaction. Denote* $\mathcal{H}_{\nu,\mu} = \mathcal{H}_{\nu\langle 1 \rangle} \otimes \mathcal{H}_{\mu\langle 2 \rangle}, \nu, \mu = 00, 01, 1$*, with inner products* $\langle \eta, \eta \rangle_{\nu,\mu}$ *and reproducing kernels* $R_{\nu,\mu} = R_{\nu\langle 1 \rangle} R_{\mu\langle 2 \rangle}$*, see Theorem 2.2.2. One may set*

$$\begin{aligned} J(\eta, \eta) &= \theta_{1,00}^{-1} \langle \eta, \eta \rangle_{1,00} + \theta_{00,1}^{-1} \langle \eta, \eta \rangle_{00,1} \\ &+ \theta_{1,01}^{-1} \langle \eta, \eta \rangle_{1,01} + \theta_{01,1}^{-1} \langle \eta, \eta \rangle_{01,1} + \theta_{1,1}^{-1} \langle \eta, \eta \rangle_{1,1}. \end{aligned}$$

The null space of $J(\eta)$ is

$$\mathcal{N}_J = \mathcal{H}_{00,00} \oplus \mathcal{H}_{01,00} \oplus \mathcal{H}_{00,01} \oplus \mathcal{H}_{01,01}.$$

As discussed in Example 3.1.1, the two dimensional η can be decomposed into four main terms: one constant term, two main effect terms, and one two-way interaction term. In Example 3.1.1, there are five effective smoothing parameters, $\lambda/\theta_{1,00}$, $\lambda/\theta_{00,1}$, $\lambda/\theta_{1,01}$, $\lambda/\theta_{01,1}$, and $\lambda/\theta_{1,1}$. Two of them, i.e., $\lambda/\theta_{1,00}$ and $\lambda/\theta_{00,1}$, are for main effects and the rest of them are for the interaction term.

Example 3.1.2 *For the tensor product cubic spline on $\{1, \cdots, K\} \times [0, 1]$, one can use the kernel $R_{0\langle 1\rangle} = 1/K$ and $R_{1\langle 1\rangle} = I_{[x_{\langle 1\rangle} = \dot{x}_{\langle 1\rangle}]} - 1/K$ on $\{1, \cdots, K\}$ and $R_{00\langle 2\rangle} = 1$, $R_{01\langle 2\rangle} = k_1(x_{\langle 2\rangle})k_1(\dot{x}_{\langle 2\rangle})$, and $R_{1\langle 2\rangle} = k_2(x_{\langle 2\rangle})k_2(\dot{x}_{\langle 2\rangle}) - k_4(x_{\langle 2\rangle} - \dot{x}_{\langle 2\rangle})$ on $[0, 1]$, where k_u's , $u = 1, 2, 4$, are scaled Bernoulli polynomials. The tensor product space in Example 3.1.1 can be analogously constructed.*

Multiple Smoothing Parameters Selection

When estimating multivariate functions through the tensor product space strategy, multiple smoothing parameters are involved, see Example 3.1.1. The multiple smoothing parameters $\boldsymbol{\lambda} = \lambda/\boldsymbol{\theta}$ control the trade-off between goals of the lack of fit of η and the roughness of η, where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_S)^t$. As reviewed in Introduction, the GCV method is a popular data-driven approach for smoothing parameters selection. Gu and Wahba (1991) proposed a modified Newton method to minimize the GCV score,

$$G(\boldsymbol{\lambda}) = \frac{n^{-1} Y^t (I - A(\boldsymbol{\lambda}))^2 Y}{(n^{-1} \operatorname{Tr}(I - A(\boldsymbol{\lambda})))^2},$$

iteratively for multiple smoothing parameters, where the smoothing matrix $A(\boldsymbol{\lambda})$ is given in Chapter 6. In particular, the method has the following two steps.

- For fixed $\boldsymbol{\theta}$, minimize the GCV score with respect to $n\lambda$
- Update $\boldsymbol{\theta}$ based on the current information of $n\lambda$.

With all smoothing parameters tunable, the above iterative algorithm takes $O(Sn^3)$ flops per iteration and needs tens of iterations to converge. The number of smoothing parameters, S, increases dramatically as the number of multi-way interactions grows. In particular, $S = d+3\binom{d}{2}$ for the two-way interaction model which truncates the decomposition in (3.3) at two-way interactions, and thus it is impractical to apply SS-ANOVA models to large samples. Even for the additive model with d smoothing parameters tunable, tens of iterations of $O(n^3)$ flops is infeasible in large samples. In order to reduce the computational load, Gu and Wahba (1991) proposed an algorithm, denoted by SKIP, to calculate starting values of $\boldsymbol{\theta}$ and used the starting values as the final estimate of $\boldsymbol{\theta}$. With the aid of SKIP, the multiple smoothing parameters selection problem is then reduced to the single smoothing parameter selection one. The algorithm, SKIP, outlined below takes $O(n^3)$ flops to estimate the starting values.

- For $\theta_\delta = (\mathrm{Tr}(R_\delta))^{-1}$, minimize the GCV score with respect to $n\lambda$, and calculate $\mathbf{c}$.
- Estimate the starting values $\theta_{\delta 0} = \theta_\delta^2 \mathbf{c}^t R_\delta \mathbf{c}$.

3.2 Asympirical Smoothing Parameters Selection Algorithm

In this section, I review the optimal smoothing parameter selection, which motivates the proposed method. The proposed smoothing parameters selection algorithm is then presented.

The Optimal Smoothing Parameter

The optimality of smoothing parameter selection can be characterized by minimizing the risk function $\mathbb{E}(L(\lambda))$, where

$$L(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\big(\eta_{n,\lambda}(x_i) - \eta(x_i)\big)^2. \tag{3.4}$$

Wahba (1975) derived the optimal smoothing parameter minimizing the risk function for smoothing periodic spline in $\mathcal{H}^{(m)}$ defined by

$$\begin{aligned}\mathcal{H}^{(m)} = \{f : f^{(\nu)} \text{ abs.cont.}, \nu = 0, 1, \cdots, m-1, f^{(m)} \in \mathcal{L}_2[0,1],\\ f^{(\nu)}(0) - f^{(\nu)}(1) = 0, \nu = 0, 1, \cdots, m-1\}.\end{aligned}$$

Suppose $\eta \in \mathcal{H}^{(2m)}$, i.e., η is very "smooth", and $\left\|\eta^{(2m)}\right\| \neq 0$, where $\|\cdot\|$ is the L_2 norm, the optimal choice of the smoothing parameter ignoring $o(1)$ is given by

$$\left(\frac{\tilde{k}_m}{4m}\frac{\sigma^2}{\|\eta^{(2m)}\|^2}\right)^{2m/(4m+1)} n^{-2m/(4m+1)}, \tag{3.5}$$

where $\tilde{k}_m = \frac{1}{\pi}\int_0^\infty \frac{dt}{(1+t^{2m})^2}$ is a constant depending on m. We rewrite the smoothing parameter in (3.5) as $Cn^{-2m/(4m+1)}$ since the first term is a constant unrelated to full sample size n. Likewise, in the subsample of size $b \to \infty$, the asymptotically optimal smoothing parameter $\breve{\lambda}$ is $Cb^{-2m/(4m+1)}$ for the same C. If we can estimate C in a subsample of size b, then the smoothing parameter $\breve{\lambda}(n/b)^{-2m/(4m+1)}$ for the full sample size n is thereby estimated. Under different "smoothness" conditions, to be defined later, the optimal smoothing parameter minimizing the risk function has the form $\tilde{C}b^{-r/(pr+1)}$ for $r > 1$ and $p \in [1, 2]$ in a subsample of size b (Wahba, 1977, 1985). For instance, we have $r = 2m$ and $p = 2$ for the above smoothing periodic spline case. Based on the same rationale described above, the smoothing parameter for the full sample is

$$\breve{\lambda}(n/b)^{-r/(pr+1)}. \tag{3.6}$$

The Asympirical Algorithm

It is infeasible to choose the optimal smoothing parameter if the true η and σ^2 are unknown. Therefore, we substitute the optimal smoothing parameter $\breve{\lambda}$ in (3.6) with the λ_g chosen by the GCV method in a subsample of size b. The detailed procedure is outlined in Algorithm 1. To make the estimated smoothing parameters more stable, I usually take multiple subsamples

Algorithm 1 Asympirical Smoothing Parameters Selection Algorithm

1: **Smoothing parameters selection in a subsample:** take a random subsample of size b from the original data and apply the GCV method to the subsample. The estimated smoothing parameters are denoted by λ_g and $\boldsymbol{\theta}_g$.
2: **Extrapolation to the full sample:** the proposed smoothing parameters $\lambda^* = \lambda_g(n/b)^{-r/(pr+1)}$ and $\boldsymbol{\theta}^* = \boldsymbol{\theta}_g$ are used to estimate the minimizer of (1.2) for the full sample of size n.

and choose the median of a group of smoothing parameters. In Algorithm 1, I assume optimal smoothing parameters share the same decreasing rate as n increases (Gu and Wahba, 1991). Since smoothing parameters $\boldsymbol{\theta}$ are used to adjust the roughness penalties imposed on different components, see Example 3.1.1, I calculate the optimal $\boldsymbol{\theta}_g$ for the subsample and perform the minimization based on the estimated $\boldsymbol{\theta}_g$ for the full sample. Further details about how to choose b, r, and p in real applications are shown in the next Section.

3.3 Theoretical Analysis

In this section, I show smoothing parameters selected by the proposed method tend to the values of the ones minimizing the risk function. The theoretical analysis provides a guide for choosing b, r, and p. Then, I present results on convergence rates of the estimator based on the proposed smoothing parameters. For simplicity, I suppress the λ's dependence on $\boldsymbol{\theta}$ and only make the λ explicit. All proofs are given in Chapter 6.

Suppose the subsample size is b, the $I - A(\lambda)$ for smoothing spline ANOVA models has the representation (see details in Chapter 6)

$$I - A(\lambda) = b\lambda Z(D + b\lambda I)^{-1} Z^t,$$

where $Z^t Z$ is a $(b-M) \times (b-M)$ identity matrix, D_{b-M} is a $(b-M) \times (b-M)$ diagonal matrix with entries $\zeta_{\nu b} > 0$. I obtain theoretical results under the "smoothness" assumption on $\eta \in \mathcal{H}_p$. The $\mathcal{H}_p$ is defined as

$$\mathcal{H}_p = \left\{ \eta : P(\eta) > 0 \text{ and } \sum_{\nu=1}^{b-M} \frac{h_{\nu b}^2/b}{(\zeta_{\nu b}/b)^p} < J_p(1 + o(1)) \right\},$$

where $(h_{1,b} \cdots h_{b-M,b})^t = Z^t \boldsymbol{\eta}$ in which $\boldsymbol{\eta} = (\eta(x_1), \cdots, \eta(x_b))^t$, J_p for $p \in [1, 2]$ is a constant independent of subsample size b. Note that I only consider the case $J(\eta) > 0$. When $J(\eta) = 0$, i.e., $\eta_{n,\lambda} \in \text{span}\{\phi_\nu\}$, both the risk function and $\mathbb{E}G(\lambda)$ are minimized for $\lambda = \infty$ (Craven and Wahba, 1978).

Theorem 3.3.1 *Suppose* $\sum \frac{h_{\nu b}^2/b}{(\zeta_{\nu b}/b)^p} < J_p$ *for some* $p \in [1, 2]$, *then for some* $r > 1$, *as* $\lambda \to 0$ *and* $b\lambda^{1/r} \to \infty$,

$$\lambda^* = \tilde{\lambda}(1 + o(1)),$$

where $\tilde{\lambda}$ *is the smoothing parameter minimizing the risk function* $\mathbb{E}L(\lambda)$ *and* $o(1) \to 0$ *as* $b \to \infty$.

I show the proposed smoothing parameter λ^* is an estimate of the minimizer of $\mathbb{E}L(\lambda)$ asymptotically. Therefore, the smoothing parameter λ^* has the order of $n^{-r/(pr+1)}$ when the full sample size is n.

In Theorem 3.3.1, one needs $b\lambda^{1/r} \to \infty$. I further assume that the λ achieves at the optimal rate $n^{-r/(pr+1)}$, and it suffices to have $b \asymp n^{1/(pr+1)+\varepsilon}$, $\forall \varepsilon > 0$. For $J(\eta) = \int_0^1 (\eta^{(2)})^2 dx$ on $[0, 1]$, we have $r = 4$, $p = 1$ when $\eta^{(2)}$ is square integrable, and $p = 2$ when $\eta^{(4)}$ is square integrable. For the tensor product cubic spline, r is typically less than 4 (Wahba, 1990;

Lin, 2000), and thus I set $r = 3$ empirically. Taking into consideration of these facts, I set $r = 3$, $p = 1$, and $\varepsilon = 0$ and use $b \propto n^{1/4}$ empirically. Note that the "smoothness" of η is indexed by p and I estimate it by an empirical way. I first take a random subsample of size B and minimize the GCV score with respect to $p \in \{1, 2\}$ by replacing the λ in the score with $\lambda_g(B/b)^{-r/(pr+1)}$. I set $B = 2b$ in the simulation and real applications. Thus the computational complexity of the proposed algorithm is of order $O(B^3)$. To reduce the computing burden of fitting SS-ANOVA models for large samples, we may implement the fast algorithm proposed by Kim and Gu (2004). In the algorithm, one first randomly selects $\breve{q}$ basis functions from n ones and then estimates the minimizer of (1.2). The algorithm requires $O(n\breve{q}^2)$ flops to estimate the minimizer for each choice of smoothing parameters. Therefore, the corresponding computational complexities of GCV and the proposed method are also reduced. Note that the complexity of the proposed method is of order $O(B\breve{q}^2)$ when the fast algorithm is applied.

I now show the convergence of η_{n,λ^*}. Let the mean squared error between the estimated and true function be $V(\eta_{n,\lambda} - \eta_0)$. To avoid interpolation, the regularization λJ needs to restrict the estimate to an effective model space. To control the bias, the effective model space needs to be increased by letting $\lambda \to 0$ as the sample size $n \to \infty$. It was shown in Gu (2013) (Chapter 9) that $(V + \lambda J)(\eta_{n,\lambda} - \eta_0) = O(n^{-1}\lambda^{-1/r} + \lambda^p)$. Then, it is trivial to show the following theorem under some conditions described in Chapter 6.

Theorem 3.3.2 *Under some conditions hold for some $p \in [1, 2]$ and $r > 1$, as $\lambda \to 0$ and $n\lambda^{2/r} \to \infty$, we have*

$$(V + \lambda J)(\eta_{n,\lambda^*} - \eta_0) = O(n^{-pr/(pr+1)}).$$

3.4 Numerical Experiments

Simulation studies and two real data applications were carried out to assess the performance of the proposed method. I used the fast algorithm proposed by Kim and Gu (2004) to reduce

the computational burden of fitting SS-ANOVA models. I randomly chose $\breve{q} = 10n^{2/9}$ basis functions from n ones in the simulation, whereas $\breve{q}$ was set to $2n^{2/3}$ for a more accurate estimation in real data. I used the same basis functions for methods in comparison. The CPU time for GCV and the proposed method, denoted by ASP, were also reported.

Simulation Study

In the simulation, I compared the proposed method, GCV, SKIP, and order based method, denoted by ORD, in terms of the mean squared error (MSE). The SKIP method was described in Section 3.1. In the order based method for comparison, I directly used $n^{-r/(pr+1)}$ as the smoothing parameter λ for sample size n and $\boldsymbol{\theta}$ were chosen by the proposed Algorithm 1. I chose the GCV method as the benchmark and reported log-transformed (natural base) relative efficacies, defined as $\log(L(\hat{\eta}, \eta)/L(\tilde{\eta}, \eta))$, where $\hat{\eta}$ is the estimator of the method for comparison and $\tilde{\eta}$ is the estimator estimated by the GCV method. The smaller of log-transformed relative efficacies indicates the better performance. If the log-transformed relative efficacies are zeros, the method has the same numerical performance compared to the GCV method. Three univariate and four multivariate functions were evaluated. The full sample size n was set to 20K, 30K, and 40K. Four values of signal to noise ratio (SNR), $1, 2, 5, 7$, defined as $\mathrm{sd}(\eta(x))/\sigma$ were used to generate the data. For each setting, 100 replicates were generated.

Single Smoothing Parameter

I simulated the data according to (1.1) using three univariate functions with different order of "smoothness" in these scenarios.

- Univariate Scenario 1:

$$\eta_{u1}(x) = \frac{1}{3}B_{20,5}(x) + \frac{1}{3}B_{12,12}(x) + \frac{1}{3}B_{7,30}(x),$$

where

$$B_{\alpha,\beta}(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1.$$

- Univariate Scenario 2:

$$\eta_{u2}(x) = 10\sin^2(2\pi x)1_{(x\le\frac{1}{2})},$$

where $1_{(x\le\frac{1}{2})}$ is an indicator function which equals 1 for $x \le \frac{1}{2}$ and 0 otherwise.

- Univariate Scenario 3:

$$\eta_{u3}(x) = 10 \times (-x + 2(x - \frac{1}{4}))1_{(x\ge\frac{1}{4})} + 2(-x + \frac{3}{4})1_{(x\ge\frac{3}{4})},$$

where $1_{(x\ge\frac{1}{4})}$ and $1_{(x\ge\frac{3}{4})}$ are two indicator functions which equal 1 when the conditions in the parentheses are satisfied and 0 otherwise.

I generated x from uniform distribution on $[0, 1]$. The generated data for three univariate functions with SNR = 1 and three true function values were shown in Figure 3.1. The log-

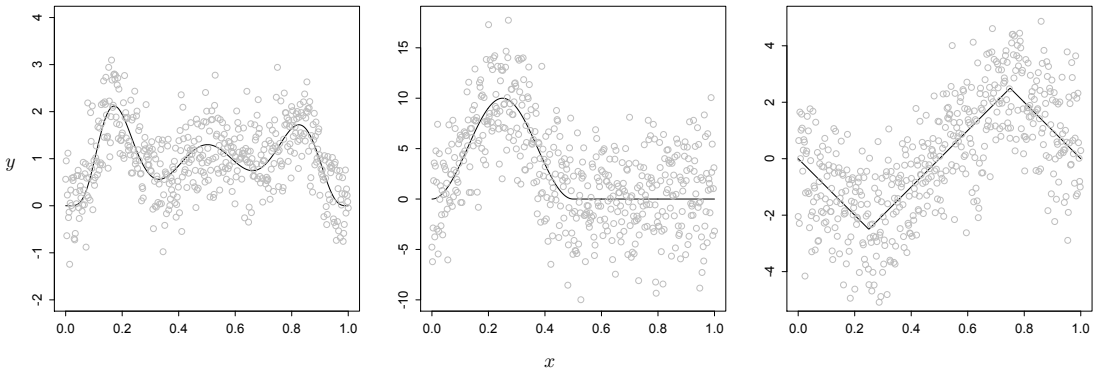

Figure 3.1: The univariate true functions (solid line) of η_{u1}, η_{u2}, and η_{u3} are shown from left to right panels respectively. The data used in the simulation are shown as circles.

transformed (natural base) relative efficacies of ASP and ORD methods for three scenarios were shown in Figure 3.2. Note the SKIP method will be reduced to the GCV method in the single smoothing parameter selection, so I do not report it. In Figure 3.2, the performance of ASP is comparable to that of GCV when SNR is low since log-transformed relative efficacies

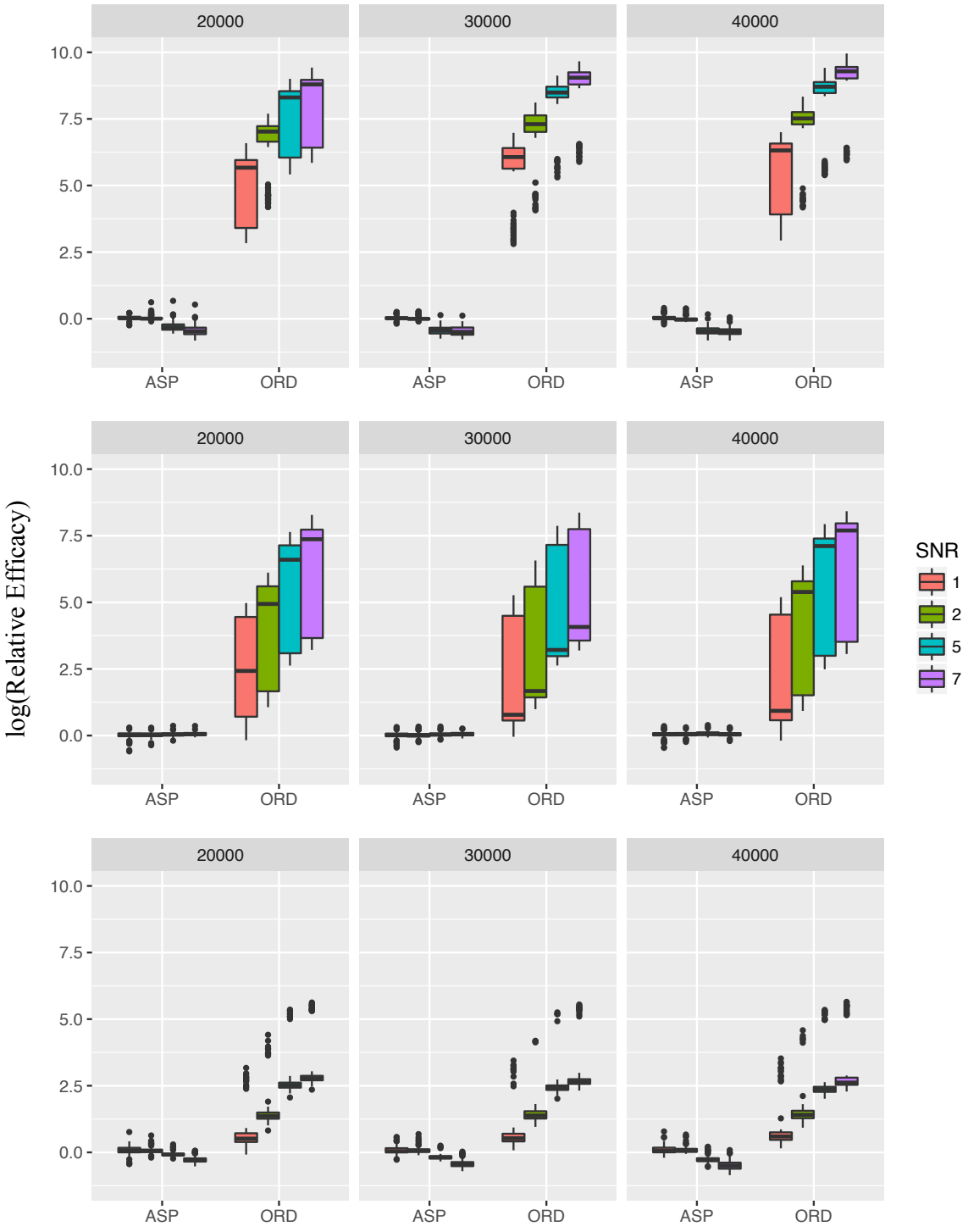

Figure 3.2: The log-transformed (natural base) relative efficacies of ASP and ORD methods over the GCV method for three Univariate Scenarios. The y-axis represents the log-transformed relative efficacies, and the x-axis represents different methods. Different SNRs are illustrated by different colors. The results of Univariate Scenario 1, 2, and 3 are shown in upper, middle, and lower panels respectively.

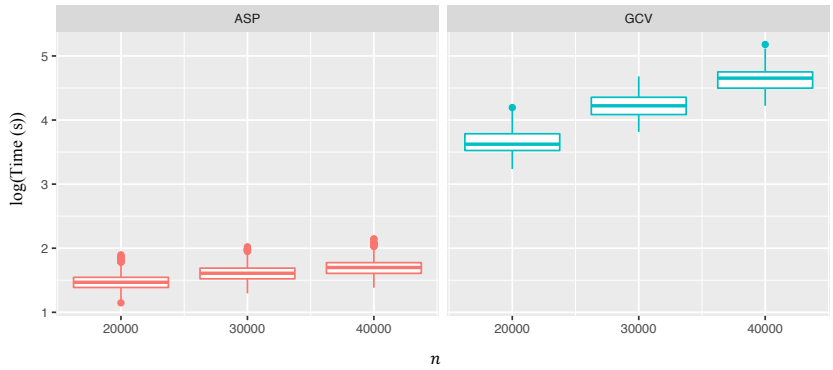

Figure 3.3: The log-transformed (natural base) CPU time of ASP (left panel) and GCV (right panel) methods for univariate functions under different sample sizes.

are close to zero. The performance of the proposed method is better than that of the GCV method as SNR increases. Such a phenomenon may be resulted from unstably estimated smoothing parameters based on subsamples when SNR is low. Even though the ORD method performs well in some scenarios, for instance, Univariate Scenario 3 under SNR = 1, it is not reliable due to the large variability in most of scenarios. In Figure 3.3, I reported the natural logarithm of CPU time for ASP and GCV methods. Typically, the CPU time of GCV is about 10 to 20 times as large as the one of ASP. The gap gets much larger as the sample size n increases.

Multiple Smoothing Parameters

I simulated the data according to (1.1) using three multivariate functions.

- Multivariate Scenario 1:

$$\eta_{m1}(x) = \frac{0.75}{\pi\sigma_{x_{\langle 1\rangle}}\sigma_{x_{\langle 2\rangle}}} e^{-\frac{(x_{\langle 1\rangle}-0.2)^2}{\sigma^2_{x_{\langle 1\rangle}}}-\frac{(x_{\langle 2\rangle}-0.3)^2}{\sigma^2_{x_{\langle 2\rangle}}}} + \frac{0.45}{\pi\sigma_{x_{\langle 1\rangle}}\sigma_{x_{\langle 2\rangle}}} e^{-\frac{(x_{\langle 1\rangle}-0.7)^2}{\sigma^2_{x_{\langle 1\rangle}}}-\frac{(x_{\langle 2\rangle}-0.8)^2}{\sigma^2_{x_{\langle 2\rangle}}}},$$

  where $\sigma_{x_{\langle 1\rangle}} = 0.3$ and $\sigma_{x_{\langle 2\rangle}} = 0.4$.

- Multivariate Scenario 2:

$$\eta_{m2}(x) = 10\sin(\pi x_{\langle 1\rangle}) + \exp(3x_{\langle 2\rangle}) + 10^6 x^{11}_{\langle 3\rangle}(1-x_{\langle 3\rangle})^6 + 10^4 x^3_{\langle 3\rangle}(1-x_{\langle 3\rangle})^{10}.$$

- Multivariate Scenario 3:

$$\eta_{m3}(x) = 10x_{\langle 2\rangle} + 10\sin(\pi(x_{\langle 3\rangle} - x_{\langle 2\rangle})) + 5\cos(2\pi(x_{\langle 1\rangle} - x_{\langle 2\rangle})).$$

- Multivariate Scenario 4:

$$\begin{aligned}\eta_{m4}(x) = \sum_{j=1}^{4} g_j(x_{\langle j\rangle}) + \sum_{j=5}^{8} 2.5 g_{j-4}(x_{\langle j\rangle}) + \sum_{j=1}^{4}\sum_{k=j+1}^{8} g_j(x_{\langle j\rangle}x_{\langle k\rangle}) \\ + g_1(x_{\langle 5\rangle}x_{\langle 6\rangle}) + g_1(x_{\langle 5\rangle}x_{\langle 7\rangle}) + g_1(x_{\langle 5\rangle}x_{\langle 8\rangle}) \\ + g_2(x_{\langle 6\rangle}x_{\langle 7\rangle}) + g_2(x_{\langle 6\rangle}x_{\langle 8\rangle}) + g_3(x_{\langle 7\rangle}x_{\langle 8\rangle}),\end{aligned}$$

  where $g_1(x) = x$, $g_2(x) = (2x-1)^2$, $g_3(x) = \sin(2\pi x)/(2-\sin(2\pi x))$, and $g_4(x) = 0.1\sin(2\pi x) + 0.2\cos(2\pi x) + 0.3\sin^2(2\pi x) + 0.4\cos^3(2\pi x) + 0.5\sin^3(2\pi x)$.

In three scenarios, x's were from uniform distribution on $[0, 1]$. The interaction model $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_{12}$ was considered for Multivariate Scenario 1, whereas, the additive model $\eta = \eta_\emptyset + \eta_1 + \eta_2 + \eta_3$ was fitted in Multivariate Scenario 2. In Multivariate Scenario 3, I considered the model $\eta = \eta_\emptyset + \eta_2 + \eta_{23} + \eta_{12}$. I further considered the high-dimensional case in Multivariate Scenario 4, and fitted the full two-way interaction model in the scenario. There are five, three, seven, and 92 effective smoothing parameters tunable for Multivariate Scenario 1, 2, 3, and 4 respectively. I showed log-transformed (natural base) relative efficacies of ASP, SKIP, and ORD methods over the GCV method in Figure 3.4. The performance of ASP is comparable with the one of GCV in Multivariate Scenario 1 and 2. In Multivariate Scenario 3, the ASP method has slightly larger relative efficacies when SNR is small, but the difference is trivial compared to the significant difference of computing time, see Figure 3.5. The SKIP method performs well when the number of smoothing parameters is small. However, it is unstable for the model with multiple interactions. In Multivariate Scenario 3, the median of relative efficacies of SKIP is more than 15, which means that it is at least 15 times as large as the MSE of GCV. In Multivariate Scenario 4, to make the GCV method

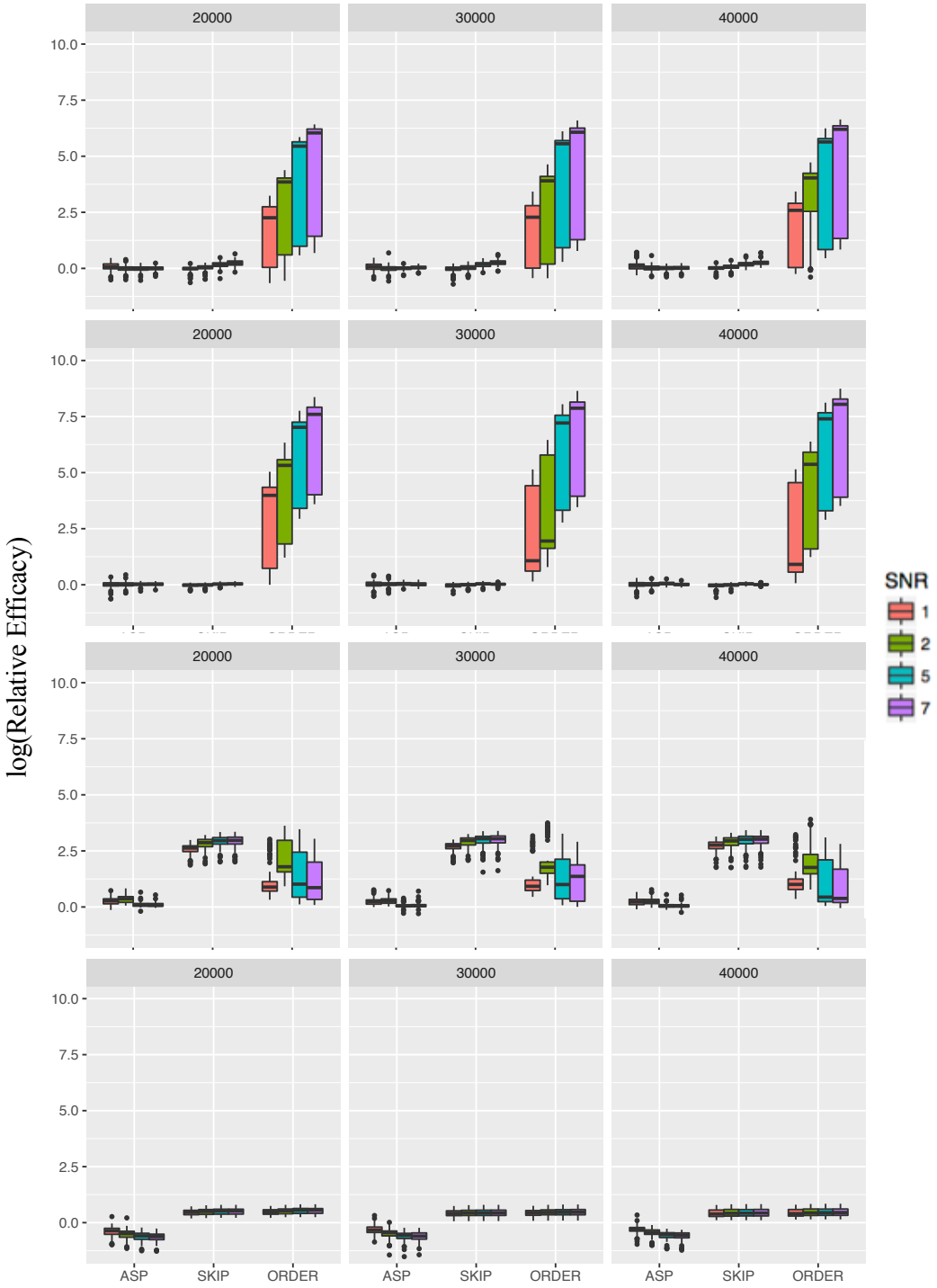

Figure 3.4: The log-transformed (natural base) relative efficacies of ASP, SKIP, and ORD methods over the GCV method for four Multivariate Scenarios. The y-axis represents log-transformed relative efficacies, and the x-axis represents different methods. Different SNRs are illustrated by different colors. The results of Multivariate Scenario 1 to 4 are shown from upper to lower panels respectively.

feasible for the high-dimensional case, I used the estimated smoothing parameters after the first iteration as the final smoothing parameters. It is expected that the ASP method performs better than the one-iteration GCV method. The relative efficacies of the SKIP method are about 2 to 3 times as large as those of the ASP method. As it is shown in univariate cases, the numerical performance of ORD is also unstable. The log-transformed (natural base) CPU time of ASP, SKIP, and GCV methods for Multivariate Scenario 3 was shown in Figure 3.5. It is expected that for each method, the runtime increases as the sample size increases. The runtimes for GCV and SKIP methods are substantially larger than those of ASP especially when the sample size is large. In particular, the CPU time of ASP is 30 times faster than GCV and 10 times faster than SKIP when the sample size is $40,000$.

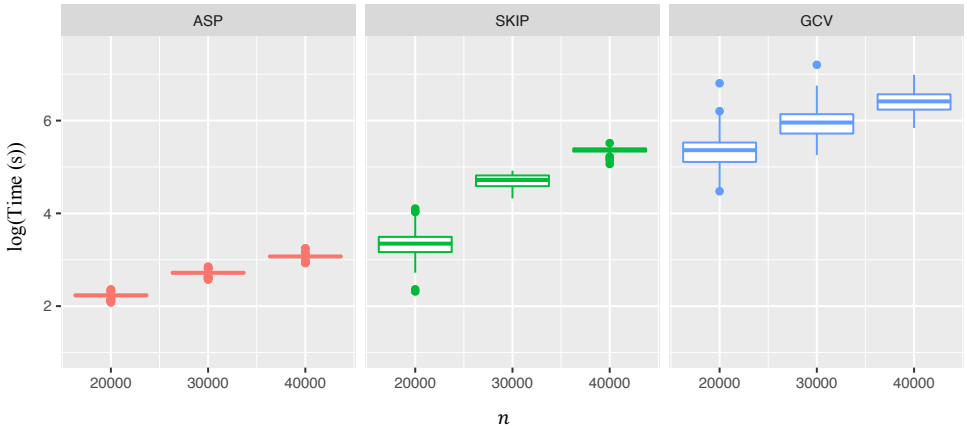

Figure 3.5: The log-transformed (natural base) CPU time of ASP (left panel), SKIP (middle panel), and GCV (right panel) methods for Multivariate Scenario 3 under different sample sizes.

Real Data Analysis

To demonstrate the potential of the proposed approach, I applied ASP and GCV methods to real data.

Beijing $PM_{2.5}$ Data

During the process of industrialization, China has been experiencing the severe environmental crisis. The particulate matter (PM), particularly the $PM_{2.5}$ of which diameter is generally 2.5 micrometers, is one of the most harmful particle pollution (Pope III et al., 2002). The dataset contains $41,757$ hourly $PM_{2.5}$ observations collected by US Embassy in Beijing and meteorological data collected at Beijing Capital International Airport in 2010-2014 (Liang et al., 2015). I used hourly $PM_{2.5}$ observations as the response and the corresponding meteorological data including time, dew point, temperature, air pressure, combined wind direction, and cumulated wind speed (m/s) were used as predictors. I fitted the cubic tensor product SS-ANOVA model to the dataset. Based on the preliminary model diagnostics (Gu, 2004), I consider the following SS-ANOVA decomposition

$$\eta = \eta_{\emptyset} + \eta_t + \eta_d + \eta_m + \eta_p + \eta_w + \eta_s + \eta_{ds},$$

where $\eta_{\emptyset}$ is a constant function, η_t, η_d, η_m, η_p, η_w, and η_s denote the main effect functions for time, dew point, temperature, air pressure, combined wind direction, and cumulated wind speed respectively, and η_{ds} denotes the interaction effect function of dew point and cumulated wind speed. There are nine effective smoothing parameters in the decomposition.

In Table 3.1, I showed the fit and predict statistics for GCV and ASP methods. The GCV method has a higher R^2 and the root MSE for fitting. I further compared the 5-fold cross-validated root MSE for prediction of these two methods by dividing the full data into 5 parts each with observations of one year. The mean and standard deviation of five root MSE results for predicting the testing data was reported. The GCV method performs slightly better than the proposed method. However, the proposed method is much faster in terms of the CPU time. Our method took about 18 seconds to calculate the smoothing parameters, whereas the GCV method took about 18.5 hours. In Figure 3.6, I presented main effect functions for time and dew point. The $PM_{2.5}$ of Beijing clearly has seasonal effect, i.e., it

is high in winter and low in summer. The high $PM_{2.5}$ in winter may relate to the winter heating in Beijing (Liang et al., 2015). Dew point is the temperature where the water vapor in the air condenses to liquid water (dew). It is referred to as frost point in winter and is highly correlated to humidity as the higher dew point means more water vapor in the air. In the right panel of Figure 3.6, the dew point does not have a significant impact on the $PM_{2.5}$ in winter (dew point < 0). As the dew point increases in summer (dew point > 10), the air humidity typically accumulates, which results in high $PM_{2.5}$.

Table 3.1: Fit and predict statistics for the SS-ANOVA model.

Method	R^2	Root Fitting MSE	Root Prediction MSE (mean)	Root Prediction MSE (sd)	CPU Time (s)
GCV	**0.514**	**63.365**	**64.108**	1.054	66917.610
ASP	0.504	63.846	64.526	**0.448**	**17.998**

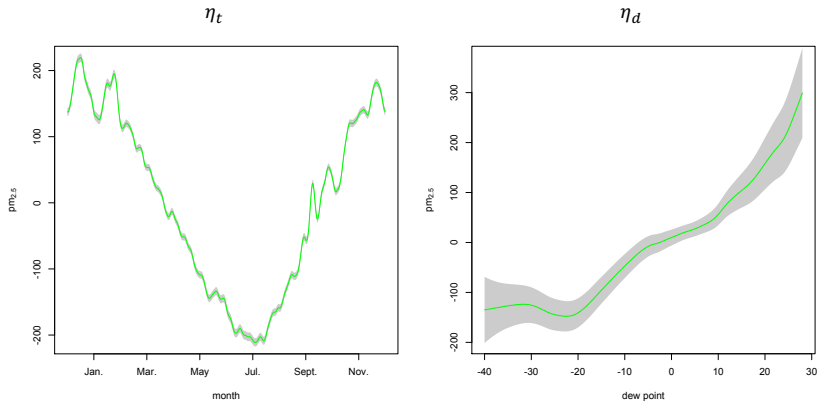

Figure 3.6: Main effect functions for time (left panel) and dew point (right panel) estimated by the ASP method.

Twitter Data

With the aid of location-based services, social media can be regarded as the major source of spatiotemporal data. The data used in the example contains $10,005,301$ tweets with location (GPS) and time stamp information (Central Standard Time, CST). These tweets were collected over five weekdays in January. To reduce the computational burden, I binned the data by using 150 bins for longitude values, 75 bins for latitude values, and 13 bins for temporal values with the bin length of two hours. The binned dataset contains $n = 52,819$

non-empty bins, i.e., data points. Followed the same procedure in Helwig et al. (2015), I used log-transformed data points, the number of tweets in each bin, as the response, and the longitude and latitude coordinates (midpoints of the bins) and time stamps (hours) as the predictors. The cubic thin-plate spline and periodic cubic smoothing spline were used to model the marginal spatial effect and temporal effect respectively. In the example, I only reported results of the additive model since it could well explain the data (Helwig et al., 2015). In the additive model, we have two effective smoothing parameters.

Table 3.2: Fit and predict statistics for the SS-ANOVA model.

Method	R^2	Root Fitting MSE	Root Prediction MSE (mean)	Root Prediction MSE (sd)	CPU Time (s)
GCV	**0.683**	**1.155**	**1.205**	0.019	1364.391
ASP	**0.683**	**1.155**	1.215	**0.018**	**17.161**

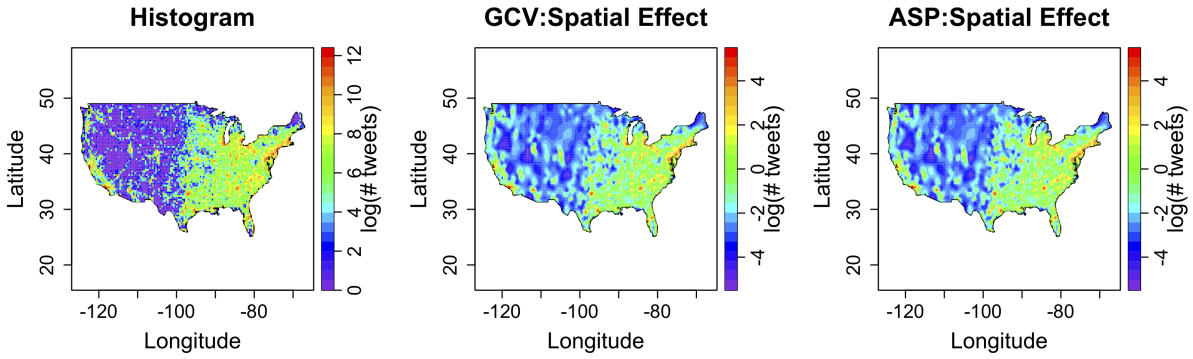

Figure 3.7: Left panel: two-dimensional histograms using 150×75 bins. Middle panel: Spatial effect function estimated by the GCV method. Right panel: Spatial effect function estimated by the ASP method.

In Figure 3.7, I compared fitted spatial effect functions with smoothing parameters estimated by GCV and ASP methods. The spatial effects of GCV and ASP methods explained the observed spatial pattern very well, which was also shown in Table 3.2. I further compared the fitting error (Root Fitting MSE) and prediction error (Root Prediction MSE) of these two smoothing parameters selection methods in Table 3.2. I divided the data into 5 parts evenly and used one part as the testing dataset. The mean and standard deviation

of five root MSE results based on the testing dataset was reported as the prediction error. The ASP method was comparable with the GCV method in terms of the fitting MSE. The GCV method improved about 0.8% performance in terms of prediction errors. However, the improvement needs extra 79 times of computing time of the proposed method.

3.5 Discussion

In this thesis, I proposed an *asympirical* (*asym*ptotic + em*pirical*) smoothing parameters selection method. The proposed method is more efficient than the GCV method since it only needs a small subsample instead of the full sample to estimate smoothing parameters. The simulation results clearly revealed numerical advantages of the proposed smoothing parameters selection method. Moreover, the theoretical analysis of the proposed smoothing parameters guaranteed they shared the same order with the ones minimizing the expected mean squared error. To study the theoretical property, we further need to assume the data are evenly observed (Wahba, 1985). This assumption is usually met in large samples. However, when the subsample size in the proposed algorithm is small, we may fail to satisfy the assumption. Under this condition, the stratified sampling based on the response values instead of uniform sampling may be used. I also showed the estimator based on the proposed smoothing parameters achieved the optimal convergence rate. The applications of our method to Beijing $PM_{2.5}$ data and Twitter data provided further evidence that the proposed method could be applied to the large and complex samples efficiently and effectively. In the proposed algorithm, I used the GCV method to select smoothing parameters for a subsample. However, the GCV method could be potentially replaced by other smoothing parameters selection methods.

Chapter 4

Optimal Penalized Function-on-Function Regression

Chapter Summary: Many scientific studies collect data where the response and predictor variables are both functions of time, location, or some other covariate. Understanding the relationship between these functional variables is a common goal in these studies. Motivated from two real-life examples, I present a function-on-function regression model that can be used to analyze such kind of functional data. The estimator of the 2D coefficient function is the optimizer of a form of penalized least squares where the penalty enforces a certain level of smoothness on the estimator. The first result is the Representer Theorem which states that the exact optimizer of the penalized least squares actually resides in a data-adaptive finite dimensional subspace although the optimization problem is defined on a function space of infinite dimensions. This theorem then allows an easy incorporation of the Gaussian quadrature into the optimization of the penalized least squares, which can be carried out through standard numerical procedures. The estimator achieves the minimax convergence rate in mean prediction under the framework of function-on-function regression. Extensive simulation studies demonstrate the numerical advantages of the proposed method over the existing ones, where a sparse functional data extension is also introduced. The proposed method is then applied to our motivating examples of the benchmark Canadian weather data and a histone regulation study. The materials of this chapter are mainly taken from Sun et al. (2018).

4.1 Penalized Function-on-Function Regression

I first introduce a simplification to model (1.3). Since model (1.3) implies that

$$Y_i(t) - \mathbb{E}Y_i(t) = \int_{I_x} \beta(t,s)\{X_i(s) - \mathbb{E}X_i(s)\}ds + \epsilon_i(t), \quad t \in I_y,$$

we may, for simplicity, only consider X and Y to be centered, i.e., $\mathbb{E}X = \mathbb{E}Y = 0$. Thus, the functional linear regression model takes the form of

$$Y_i(t) = \int_{I_x} \beta\big(t,s\big)X_i(s)ds + \epsilon_i(t), \quad t \in I_y. \tag{4.1}$$

The Representer Theorem

Assume that the unknown β resides in a RKHS $\mathcal{H}(R)$ with the reproducing kernel $R : I \times I \to \mathbb{R}$, where $I = I_y \times I_x$. The estimate $\hat{\beta}_n$ can be obtained by minimizing the following penalized least squares functional

$$\frac{1}{n}\sum_{i=1}^{n}\int_{I_y}\Big\{Y_i(t) - \int_{I_x}\beta(t,s)X_i(s)ds\Big\}^2 dt + \lambda J(\beta) \tag{4.2}$$

with respect to $\beta \in \mathcal{H}(R)$, where the sum of integrated squared errors represents the goodness-of-fit, J is a roughness penalty on β, and $\lambda > 0$ is the smoothing parameter balancing the trade-off. We now establish a Representer Theorem stating that $\hat{\beta}_n$ actually resides in a finite dimensional subspace of $\mathcal{H}(R)$. This result generalizes Theorem 1 in Yuan and Cai (2010) and facilitates the computation by reducing an infinite dimensional optimization problem to a finite dimensional one.

Note that the penalty functional J is a squared semi-norm on $\mathcal{H}(R)$. Its null space $\mathcal{H}_0 = \{\beta \in \mathcal{H}(R) : J(\beta) = 0\}$ is a finite-dimensional linear subspace of $\mathcal{H}(R)$. Denote by $\mathcal{H}_1$ its orthogonal complement in $\mathcal{H}(R)$ such that $\mathcal{H}(R) = \mathcal{H}_0 \oplus \mathcal{H}_1$. For any $\beta \in \mathcal{H}(R)$, there exists

a unique decomposition $\beta = \beta_0 + \beta_1$ where $\beta_0 \in \mathcal{H}_0$ and $\beta_1 \in \mathcal{H}_1$. Let $R_0(\cdot,\cdot)$ and $R_1(\cdot,\cdot)$ be the corresponding reproducing kernels of $\mathcal{H}_0$ and $\mathcal{H}_1$. Then R_0 and R_1 are both nonnegative definite operators on L_2, and $R = R_0 + R_1$. In fact the penalty term $J(\beta) = \|\beta\|^2_{R_1} = \|\beta_1\|^2_{R_1}$. By the theory of RKHS, $\mathcal{H}(R)$ has a tensor product decomposition $\mathcal{H}(R) = \mathcal{H}_y(R_y) \otimes \mathcal{H}_x(R_x)$. Here $\mathcal{H}_y(R_y)$ is the RKHS with a reproducing kernel $R_y : I_y \times I_y \to \mathbb{R}$, and $\mathcal{H}_x(R_x)$ is the RKHS with a reproducing kernel $R_x : I_x \times I_x \to \mathbb{R}$. For the reproducing kernels, we have $R(t,s) = R_y(t)R_x(s)$. Note that the functions in $\mathcal{H}_y(R_y)$ and $\mathcal{H}_x(R_x)$ are univariate and defined respectively on I_y and I_x. Similar to the decomposition of $\mathcal{H}(R)$ and R, we have the tensor sum decompositions of the marginal subspaces $\mathcal{H}_y(R_y) = \mathcal{H}_{0y} \oplus \mathcal{H}_{1y}$ and $\mathcal{H}_x(R_x) = \mathcal{H}_{0x} \oplus \mathcal{H}_{1x}$, and the orthogonal decompositions of the marginal reproducing kernels $R_y = R_{0y} + R_{1y}$ and $R_x = R_{0x} + R_{1x}$. Here R_* is a reproducing kernel on $\mathcal{H}_*$ with $*$ running through the index set $\{0y, 1y, 0x, 1x\}$.

Upon piecing the marginal decomposition parts back to the tensor product space, we obtain $\mathcal{H}_0 = \mathcal{H}_{0y} \otimes \mathcal{H}_{0x}$ and $\mathcal{H}_1 = (\mathcal{H}_{0y} \otimes \mathcal{H}_{1x}) \oplus (\mathcal{H}_{1y} \otimes \mathcal{H}_{0x}) \oplus (\mathcal{H}_{1y} \otimes \mathcal{H}_{1x})$. Correspondingly, the reproducing kernels satisfy that

$$R_0((t_1,s_1),(t_2,s_2)) = R_{0y}(t_1,t_2)R_{0x}(s_1,s_2),$$
$$R_1((t_1,s_1),(t_2,s_2)) = R_{0y}(t_1,t_2)R_{1x}(s_1,s_2) + R_{1y}(t_1,t_2)R_{0x}(s_1,s_2) + R_{1y}(t_1,t_2)R_{1x}(s_1,s_2).$$

Let $N_y = \dim(\mathcal{H}_{0y})$ and $N_x = \dim(\mathcal{H}_{0x})$. Denote by $\{\psi_{k,y} : k = 1,\ldots,N_y\}$ and $\{\psi_{l,x} : l = 1,\ldots,N_x\}$ respectively the basis functions of $\mathcal{H}_{0y}$ and $\mathcal{H}_{0x}$. With some abuse of notation, define $(R_{1y}g)(\cdot) = \int_{I_y} R_{1y}(\cdot,t)g(t)dt$ and $(R_{1x}f)(\cdot) = \int_{I_x} R_{1x}(\cdot,s)f(s)ds$. Now we can state the Representer Theorem as follows with its proof collected in Chapter 6.

Theorem 4.1.1 *Let $\hat{\beta}_n$ be the minimizer of (4.2) in $\mathcal{H}$. Then $\hat{\beta}_n$ resides in the subspace of*

functions of the form

$$\beta(t,s) = \Big\{ \sum_{k=1}^{N_y} d_{k,\beta_y}\psi_{k,y}(t) + \sum_{i=1}^{n} c_{i,\beta_y}(R_{1y}Y_i)(t)\Big\}\Big\{ \sum_{l=1}^{N_x} d_{l,\beta_x}\psi_{l,x}(s) + \sum_{j=1}^{n} c_{j,\beta_x}(R_{1x}X_j)(s)\Big\}$$
$$= \Big\{ d_{\beta_y}^t\psi_y(t) + c_{\beta_y}^t(R_{1y}Y)(t)\Big\}\Big\{ d_{\beta_x}^t\psi_x(s) + c_{\beta_x}^t(R_{1x}X)(s)\Big\}, \tag{4.3}$$

where $d_{\beta_y} = (d_{1,\beta_y},\ldots,d_{N_y,\beta_y})^t$, $c_{\beta_y} = (c_{1,\beta_y},\ldots,c_{n,\beta_y})^t$, $d_{\beta_x} = (d_{1,\beta_x},\ldots,d_{N_x,\beta_x})^t$ *and* $c_{\beta_x} = (c_{1,\beta_x},\ldots,c_{n,\beta_x})^t$ *are some coefficient vectors, and* $\psi_x, \psi_y, R_{1y}Y$ *and* $R_{1x}X$ *are vectors of functions.*

For the purpose of illustration, I give a detailed example below.

Example 4.1.1 *Consider the case of tensor product cubic splines with* $I_y = I_x = [0,1]$. *The marginal spaces* $\mathcal{H}_y(R_y) = \mathcal{H}_x(R_x) = \{g : \int_0^1 (g'')^2 < \infty\}$ *with the inner product*

$$\langle f, g\rangle_{\mathcal{H}_y} = \Big(\int_0^1 f \int_0^1 g + \int_0^1 f' \int_0^1 g' \Big) + \int_0^1 f''g''dt.$$

The marginal space $\mathcal{H}_y(R_y)$ *can be further decomposed into the tensor sum of* $\mathcal{H}_{0y} = \{g : g'' = 0\}$ *and* $\mathcal{H}_{1y} = \{g : \int_0^1 g = \int_0^1 g' = 0, \int_0^1 (g'')^2 < \infty\}$. *The reproducing kernel* R_y *is the orthogonal sum of* $R_{0y}(t_1,t_2) = 1 + r_1(t_1)r_1(t_2)$ *and* $R_{1y}(t_1,t_2) = r_2(t_1)r_2(t_2) - r_4(|t_1 - t_2|)$, *where* $r_\nu(t) = B_\nu(t)/\nu!$ *is a scaled version of the Bernoulli polynomial* B_ν. *The space* $\mathcal{H}_{0y}$ *has a dimension of* $N_y = 2$ *and a set of basis functions* $\{1, r_1(t)\}$.

The function space $\mathcal{H}$ *is defined as* $\mathcal{H} = \{\beta : J(\beta) < \infty\}$ *with the reproducing kernel* $R(t,s) = R_y(t)R_x(s)$ *and the penalty functional*

$$J(\beta) = \int_0^1 \Big[\Big\{ \int_0^1 \frac{\partial^2}{\partial s^2}\beta(t,s)dt\Big\}^2 + \Big\{ \int_0^1 \frac{\partial^3}{\partial t\partial s^2}\beta(t,s)dt\Big\}^2\Big]ds$$
$$+ \int_0^1 \Big[\Big\{ \int_0^1 \frac{\partial^2}{\partial t^2}\beta(t,s)ds\Big\}^2 + \Big\{ \int_0^1 \frac{\partial^3}{\partial t^2\partial s}\beta(t,s)ds\Big\}^2\Big]dt + \int_0^1\int_0^1 \Big\{\frac{\partial^4}{\partial t^2\partial s^2}\beta(t,s)\Big\}^2 dtds$$

We have $\mathcal{H} = \mathcal{H}_y(R_y) \otimes \mathcal{H}_x(R_x)$ *and* $R = R_yR_x$; *see, e.g., Chapter 2 of Gu (2013).*

Estimation Algorithm

To introduce the computational algorithm, we first need some simplification of notation. Let $N = N_y N_x$ and $L = n(N_y + N_x + n)$. I rewrite the functions spanning the subspace in Theorem 4.1.1 as $\psi_1(t,s) = \psi_{1,y}(t)\psi_{1,x}(s)$, $\cdots$, $\psi_N(t,s) = \psi_{N_y,y}(t)\psi_{N_x,x}(s)$ and $\xi_1(t,s) = \psi_{1,y}(t)(R_{1x}X_1)(s), \cdots, \xi_L(t,s) = (R_{1y}Y_n)(t)(R_{1x}X_n)(s)$. Thus a function in this subspace has the form $\beta(t,s) = \mathbf{d}^t\psi(t,s)+\mathbf{c}^t\xi(t,s)$ for some coefficient vectors $\mathbf{d}$, $\mathbf{c}$ and vectors of functions $\psi(t,s)$, $\xi(t,s)$. To solve (4.2), I choose Gaussian quadrature with the Gauss-Legendre rule to calculate the integrals. Consider the Gaussian quadrature evaluation of an integral on I_y with knots $\{t_1, \cdots, t_{N_q}\}$ and weights $\{\alpha_1, \cdots, \alpha_{N_q}\}$ such that $\int_{I_y} f(t)dt = \sum_{j=1}^{N_q} \alpha_j f(t_j)$. Let W be the diagonal matrix with $\alpha_1, \cdots, \alpha_{N_q}$ repeating n times on the diagonal. Then the estimation of β in (4.2) reduces to the minimization of

$$(Y_w - S_w\mathbf{d} - K_w\mathbf{c})^t(Y_w - S_w\mathbf{d} - K_w\mathbf{c}) + n\lambda\mathbf{c}^tQ\mathbf{c} \tag{4.4}$$

with respect to $\mathbf{d}$ and $\mathbf{c}$, where $Y_w = W^{1/2}Y$ with $Y = (Y_1(t_1), \ldots, Y_1(t_{N_q}), \ldots, Y_n(t_1), \ldots, Y_n(t_{N_q}))^t$, $S_w = W^{1/2}S$ with S being an $nN_q \times N$ matrix with the $((i-1)N_q+j, \nu)$th entry $\int_{I_x} \psi_\nu(t_j,s)X_i(s)ds$, $K_w = W^{1/2}K$ with K being an $nN_q \times L$ matrix with the $((i-1)N_q+j, k)$th entry $\int_{I_x} \xi_k(t_j,s)X_i(s)ds$, and Q is a $L \times L$ matrix with the (i,j)th entry $\langle \xi_i, \xi_j \rangle_{\mathcal{H}_1}$. Let $Q_x = \left[\int_0^1 \int_0^1 X_i(u)R(u,v)X_j(v)dudv\right]_{i,j=1}^n$, $Q_y = \left[\int_0^1 \int_0^1 Y_i(u)R(u,v)Y_j(v)dudv\right]_{i,j=1}^n$, and $Q_{xy} = Q_x \otimes Q_y$, we have $Q = \mathrm{diag}(Q_x, Q_x, Q_y, Q_y, Q_{xy})$.

I then utilize standard numerical linear algebra procedures such as the Cholesky decomposition with pivoting and forward and back substitutions, to calculate $\mathbf{c}$ and $\mathbf{d}$ in (4.4) (Gu, 2013, Section 3.5). To choose the smoothing parameter λ in (4.4), a modified generalized cross-validation (GCV) score (Craven and Wahba, 1978),

$$G(\lambda) = \frac{(nN_q)^{-1}Y_w^t(I - A(\lambda))^2Y_w}{\{(nN_q)^{-1}tr(I - \alpha A(\lambda))\}^2} \tag{4.5}$$

is implemented, where $\alpha > 1$ is a fudge factor curbing undersmoothing (Kim and Gu, 2004)

and $A(\lambda)$ is the smoothing matrix bridging the prediction $\hat{Y}_w$ and the observation Y_w as $\hat{Y}_w = A(\lambda)Y_w$, similar to the hat matrix in a general linear model.

4.2 Optimal Mean Prediction Risk

We are interested in the estimation of coefficient function β and mean prediction, that is, to recover the functional $\eta_\beta(X, \cdot) = \int_{I_x} \beta(\cdot, s)X(s)ds$ based on the training sample (X_i, Y_i), $i = 1, \ldots, n$. Let $\hat{\beta}_n(t, s)$ be an estimate of $\beta(t, s)$. Suppose (X_{n+1}, Y_{n+1}) is a new observation that has the same distribution as and is also independent of (X_i, Y_i), $i = 1, \ldots, n$. Then the prediction accuracy can be naturally measured by the excess risk

$$
\begin{aligned}
&\mathfrak{R}_n(\hat{\beta}_n) \\
&= \int_{I_y} \left[\mathbb{E}^* \Big\{ Y_{n+1}(t) - \int_{I_x} \hat{\beta}_n(t, s) X_{n+1}(s) ds \Big\}^2 - \mathbb{E}^* \Big\{ Y_{n+1}(t) - \int_{I_x} \beta(t, s) X_{n+1}(s) ds \Big\}^2 \right] dt \\
&= \int_{I_y} \mathbb{E}^* \Big\{ \eta_{\hat{\beta}_n}(X_{n+1}, t) - \eta_\beta(X_{n+1}, t) \Big\}^2 dt
\end{aligned}
$$

where $\mathbb{E}^*$ represents the expectation taken over (X_{n+1}, Y_{n+1}) only. We shall study the convergence rate of $\mathfrak{R}_n$ as the sample size n increases.

This section collects two theorems whose combination indicates that the estimator achieves the optimal minimax convergence rate in mean prediction. We first establish the minimax lower bound for the convergence rate of the excess risk $\mathfrak{R}_n$. There is a one-to-one relationship between R and $\mathcal{H}(R)$ which is a linear functional space endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}(R)}$ such that

$$
\beta(t, s) = \Big\langle R\big((t, s), \cdot\big), \beta \Big\rangle_{\mathcal{H}(R)}, \quad \text{for any } \beta \in \mathcal{H}(R).
$$

The kernel R can also be treated as an integral operator such that

$$R(\beta)(\cdot) = \Big\langle R\big((t,s),\cdot\big), \beta \Big\rangle_{L_2} = \int\int_I R((t,s),\cdot)\beta(t,s)dtds.$$

It follows from the spectral theorem that there exist a set of orthonormal eigenfunctions $\{\zeta_k : k \geq 1\}$ and a sequence of eigenvalues $\kappa_1 \geq \kappa_2 \geq \cdots > 0$ such that

$$R((t_1,s_1),(t_2,s_2)) = \sum_{k=1}^{\infty} \kappa_k \zeta_k(t_1,s_1)\zeta_k(t_2,s_2), \qquad R(\zeta_k) = \kappa_k\zeta_k, \quad k = 1,2,\ldots.$$

Denote $R^{1/2}\big((t_1,s_1),(t_2,s_2))\big) = \sum_{k=1}^{\infty} \kappa_k^{1/2}\zeta_k(t_1,s_1)\zeta_k(t_2,s_2)$. Let $C(t,s) = \mathrm{cov}\big(X(t),X(s)\big)$ be the covariance kernel of X. Define a new kernel Π such that

$$\Pi\big((t_1,s_1),(t_2,s_2)\big) = \int\int\int_{I_x\times I_x\times I_y} R^{1/2}\big((t_1,s_1),(z,u)\big)\ C(u,v)\ R^{1/2}\big((t_2,s_2),(z,v)\big)dudvdz. \tag{4.6}$$

Let $\rho_1 \geq \rho_2 \geq \cdots > 0$ be the eigenvalues of Π and $\{\phi_j : j \geq 1\}$ be the corresponding eigenfunctions. Therefore,

$$\Pi\big((t_1,s_1),(t_2,s_2)\big) = \sum_{k=1}^{\infty} \rho_k\phi_k(t_1,s_1)\phi_k(t_2,s_2), \quad \forall\ (t_1,s_1),(t_2,s_2) \in I_y \times I_x.$$

Theorem 4.2.1 *Assume that for any $\beta \in L_2([0,1]^2)$*

$$\int \mathbb{E}\Big(\int \beta(t,s)X(s)dt\Big)^4 dt \leq c\int\Big(\mathbb{E}\Big(\int\beta(t,s)X(s)ds\Big)^2\Big)^2 dt \tag{4.7}$$

for a positive constant c. Suppose that the eigenvalues $\{\rho_k : k \geq 1\}$ of the kernel Π in (4.6) satisfy $\rho_k \asymp k^{-2r}$ for some constant $0 < r < \infty$. Then,

$$\lim_{A\to\infty}\lim_{n\to\infty}\sup_{\beta\in\mathcal{H}(R)} \mathbb{P}\Big\{\mathfrak{R}_n \geq An^{-\frac{2r}{2r+1}}\Big\} = 0, \tag{4.8}$$

when λ is of order $n^{-2r/(2r+1)}$.

Theorem 4.2.1 indicates that the convergence rate is determined by the decay rate of the eigenvalues of this new operator Π, which is jointly determined by both reproducing kernel R and the covariance kernel C as well as the alignment between R and C in a complicated way. This result has not been reported in the literature before. A close and related result is from Yuan and Cai (2010) who studied an optimal prediction risk for functional linear models, where the optimal rate depends on the decay rate of the eigenvalues of $R^{1/2}CR^{1/2}$. It is interesting to see, on the other hand, whether the convergence rate of $\hat{\beta}_n$ in Theorem 4.2.1 is optimal. In the following, a minimax lower bound for the risk is derived.

Theorem 4.2.2 *Let r be as in Theorem 4.2.1. Then the excess prediction risk satisfies*

$$\lim_{c\to 0}\lim_{n\to\infty}\inf_{\tilde{\eta}}\sup_{\beta\in\mathcal{H}(R)}\mathbb{P}\Big(\mathfrak{R}_n \geq cn^{-\frac{2r}{2r+1}}\Big) = 1, \tag{4.9}$$

where the infimum is taken over all possible predictors $\tilde{\eta}$ based on $\{(X_i, Y_i) : i = 1, \ldots, n\}$.

Theorem 4.2.2 shows that the minimax lower bound of the convergence rate for the prediction risk is $n^{-2r/2r+1}$, which is determined by r and the decay rate of the eigenvalues of Π. This rate is then achieved by the proposed penalized estimator, and therefore the estimator is rate-optimal.

4.3 Numerical Experiments

I compared the proposed optimal penalized function-on-function regression (OPFFR) method with existing function-on-function linear regression models under two different designs. In a dense design, each curve was densely sampled at regularly-spaced common time points. I compared the OPFFR with two existing models. In a sparse design, each curve was irregularly and sparsely sampled at possibly different time points. I extended the OPFFR to this design by adding an extra pre-smoothing step and compared it with the FPCA model. In the first model (Ramsay and Silverman, 2005) for comparison, the coefficient function

is estimated by penalizing its B-spline basis function expansion. This approach does not have the optimal mean prediction property and partially implemented in the `fda` package of R (`linmod` function) for the case of a fixed smoothing parameter. I shall add a search on the grid $10^{(-2:0.4:2)}$ for smoothing parameter selection to their implementation and denote this augmented approach by FDA. The coefficient function is represented in terms of 10 basis functions each for the t and s directions. The second model for comparison was the functional principal component analysis (hence denoted by FPCA) approach proposed by Yao et al. (2005b). The coefficient function is represented in terms of the leading functional principal components. This is implemented in the MatLab package `PACE` (`FPCreg` function) maintained by the UC-Davis research group. The Akaike information criterion (AIC) and fraction of variance explained (FVE) criterion were used to select the number of principal components for predictor and response respectively. The cutoff value for FVE was 0.9. The 'regular' parameter was set to 2 for the dense design and 0 for the sparse design. No binning was performed.

Simulation Study

Dense Design

I simulated data according to model (4.1) with three scenarios.

- Scenario 1: The predictor functions are $X_i(s) = \sum_{k=1}^{50}(-1)^{(k+1)}k^{-1}Z_{ik}\vartheta_1(s,k)$, where Z_{ik} is from the uniform distribution $U(-\sqrt{3},\sqrt{3})$, and $\vartheta_1(s,k) = 1$ if $k = 1$ and $\sqrt{2}\cos((k-1)\pi s)$ otherwise. The coefficient function $\beta(t,s) = e^{-(t+s)}$ is the exponential function of t and s.

- Scenario 2: The predictor functions $X_i(s)$ are the same as those in Scenario 1 and the coefficient function $\beta(t,s) = 4\sum_{k=1}^{50}(-1)^{(k+1)}k^{-2}\vartheta_1(t,k)\vartheta_1(s,k)$.

- Scenario 3: The predictor functions $X_i(s)$ are generated as $X_i(s) = \sum_{k=1}^{3}(-1)^{(k+1)}k^{-1}Z_{ik}\vartheta_2(s,k)$, where $\vartheta_2(s,k) = 1$ if $k = 3$ and $\sqrt{2}\cos(k\pi s)$ otherwise. The coefficient function

$$\beta(t,s) = 4\sum_{k=1}^{3}(-1)^{(k+1)}k^{-2}\vartheta_2(t,k)\vartheta_2(s,k).$$

For each simulation scenario, I generated $n = 30$ samples, each with 20 time points on the interval $(0, 1)$. The random errors $\epsilon(t)$ were from a normal distribution with a constant variance σ^2. The value of σ was adjusted to deliver three levels of signal-to-noise ratio (SNR= 0.5, 5, and 10) in each scenario. To assess the mean prediction accuracy, I generated an additional $n^* = 30$ predictor curves $\tilde{X}$ and computed the mean integrated squared error $\text{MISE} = 1/n^* \sum_{i=1}^{n^*} \int_0^1 (\eta_{\hat{\beta}}(\tilde{X}_i, t) - \eta_{\beta}(\tilde{X}_i, t))^2 dt$, where $\hat{\beta}$ was the estimator obtained from the training data. We had 100 runs for each combination of scenario and SNR.

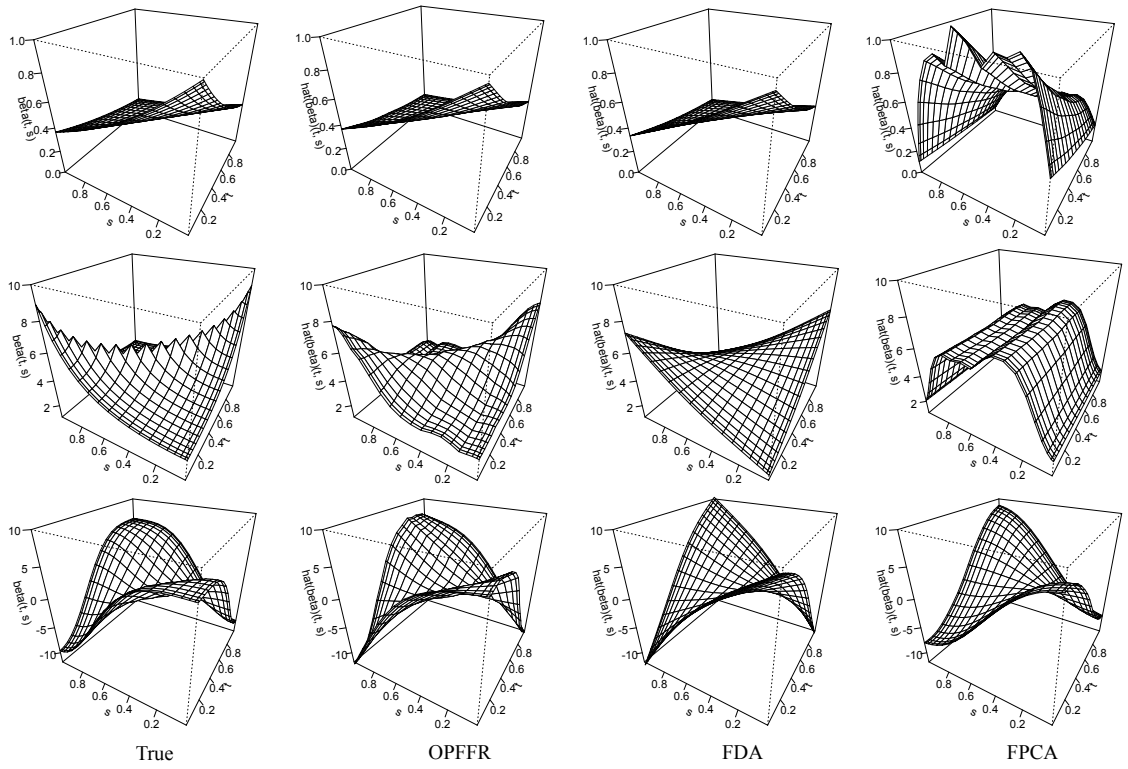

Figure 4.1: Perspective plots of the true $\beta(t, s)$ in three scenarios, and their respective estimates by the OPFFR, FDA, and FPCA methods when SNR= 10.

I applied the OPFFR, FDA and FPCA methods to the simulated datasets. Figure 4.1 displayed the perspective plots of the true coefficient functions in the three scenarios as well as their respective estimates for a single run with SNR= 10. In the first two scenarios, both OPFFR and FDA did a decent job in recovering the true coefficient function although

the FDA estimates were slightly oversmoothed. In both scenarios the FPCA estimates clearly suffered since the true coefficient function could not be effectively represented by the eigen-functions of the predictor processes. Figure 4.2 gave the summary reports of performances in terms of MISEs based on 100 runs. When the signal to noise ratio is low, the OPFFR and FDA approaches had comparable performances. But when the signal to noise ratio increases, OPFFR showed clear advantage against FDA. The FPCA method failed to deliver competitive performance against the other two methods in all the settings due to its restrictive requirement of the effective representation of the coefficient function.

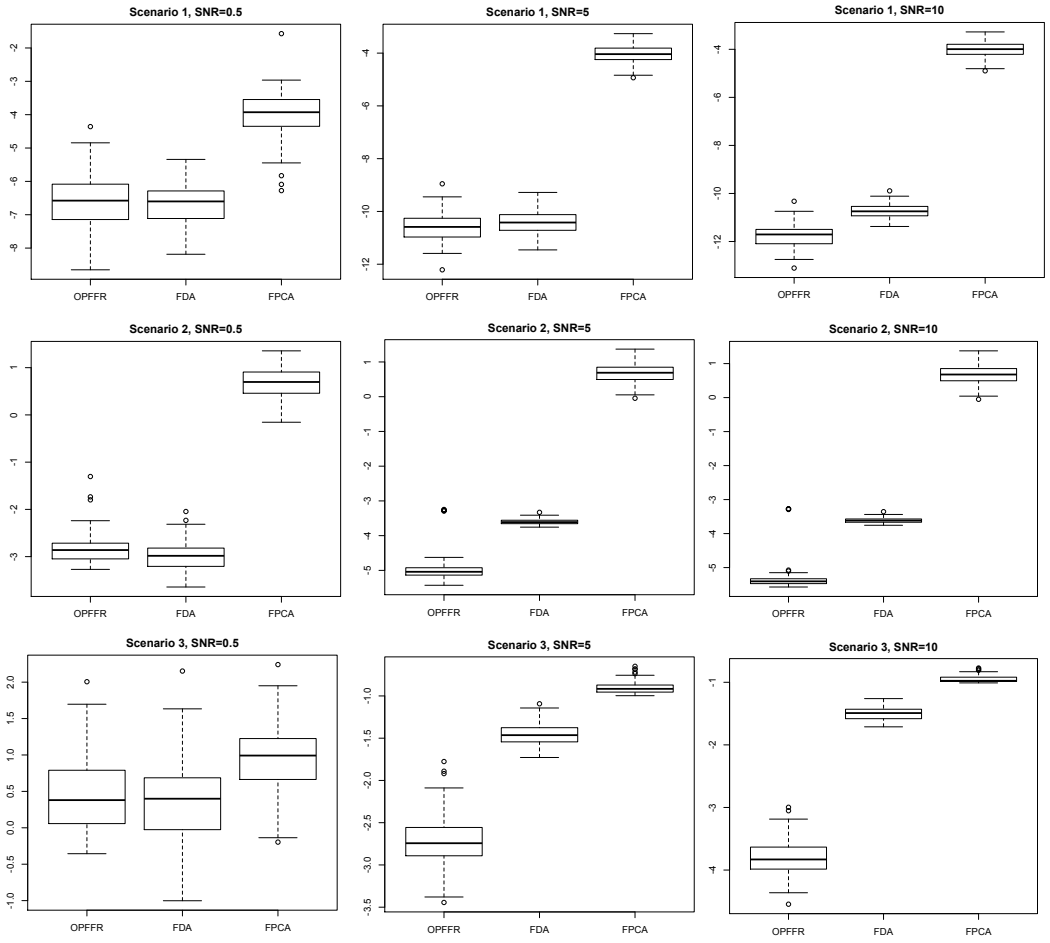

Figure 4.2: Boxplots of log(MISE) for three scenarios under three signal-to-noise ratios (SNR= 0.5, 5, 10), based on 100 simulation runs. OPFFR is the proposed approach.

Sparse Design

In this section, I compared the performance of the proposed OPFFR method and the FPCA method regarding prediction error on sparsely, irregularly, and noisily observed functional data. To extend the proposed method to sparsely and noisily observed data, I first applied the principal-component-analysis-through-conditional-expectation (PACE) method in Yao et al. (2005a) to the sparse functional data. Then we obtained a dense version of functional data by computing the PACE-fitted response and predictor functions at 50 selected time points for each curve. I applied the OPFFR method to these densely generated data and called this sparse extension to the OPFFR by the OPFFR-S method. The original OPFFR method, FPCA and OPFFR-S methods were all applied to the simulated data for comparison.

I first generated $n = 200$ samples for both response and predictor functions in Scenario 3, each with 50 time points on interval $(0, 1)$. To obtain different sparsity levels, I then randomly chose 5, 10 and 15 time points from the 50 ones for each curve independently. Normally distributed random errors were added to functional response and predictor with the SNR set to 10 in generating each pair of noisy response and predictor. The mean integrated squared error (MISE) was calculated based on additional $n^* = 50$ predictor curves without random noises. Figure 4.3 displayed the perspective plots of the true coefficient functions in the sparse scenario as well as their respective estimates for a single run with 10 sampled time points per curve. The OPFFR-S method and FPCA performed well in estimating the coefficient function. The estimate recovered by the original OPFFR method was a little oversmoothed. In Figure 4.4, the performance in terms of MISEs based on 100 runs was compared. The OPFFR-S method always had the best prediction performances at all the three sparsity levels. When the sparsity level was high (5 time points per curve), the original OPFFR method had a worse prediction performance than the FPCA. However, its prediction performance quickly picked up as the data became denser. When the sparsity level was 15 time points per curve, it actually delivered a better prediction performance than the FPCA. Such an interesting phenomenon was referred to as the "phase transistion" (Cai and Yuan,

2011; Wang et al., 2016).

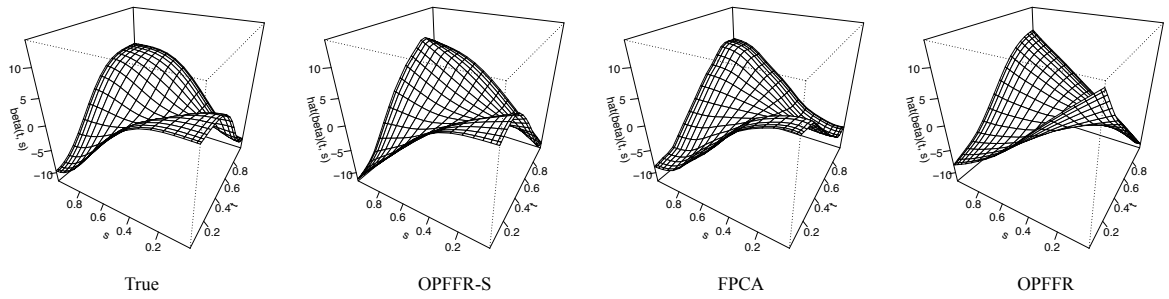

Figure 4.3: Perspective plots of the true $\beta(t, s)$ in the sparse scenario, and their respective estimates by the OPFFR-S, FPCA, and OPFFR methods when the number of randomly selected time points is ten.

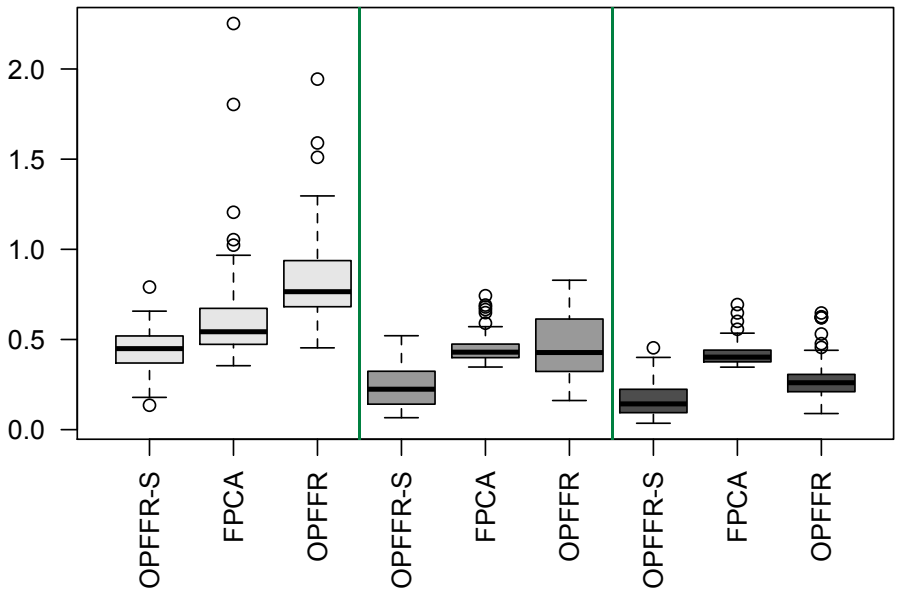

Figure 4.4: Boxplots of MISEs for the sparse scenario under three different sparsity levels, based on 100 simulation runs. The boxplots with different grayscale shades from left to right respectively represent the sparsity levels of 5, 10 and 15 time points per curve.

Real Data Analysis

I analyzed two real example in this section. I showed that our method had the numerical advantage over other approaches in terms of prediction accuracy in the analysis of the Canadian weather and histone regulation data. The results in the Canadian weather data, a dense

design case, and the histone regulation data, a sparse design case, echoed with our findings in the simulation study. The smoothing parameters used in FDA for Canadian weather data were taken from the example codes in Ramsay et al. (2009) and seven basis functions were used for the t and s directions respectively. In the histone regulation data I selected the smoothing parameter for FDA by a grid search on $10^{(-5:1:5)}$ and used six basis functions each for the t and s directions. For the FPCA method, the 'regular' parameter was set to 2 for the Canadian weather data and 0 for the histone regulation data. The other parameters for FDA and FPCA approaches were the same as those used in the simulation study.

Canadian Weather Data

I first look at the Canadian weather data (Ramsay and Silverman, 2005), a benchmark dataset in functional data analysis. The main goal is to predict the log daily precipitation profile based on the daily temperature profile for a geographic location in Canada. The daily temperature and precipitation data averaged over 1960 to 1994 were recorded at 35 locations in Canada. I compared OPFFR with FDA and FPCA in terms of prediction performance defined by integrated squared error (ISE) $\int_0^{365}(Y_i(t) - \eta_{\hat{\beta}_{-i}}(X_i, t))^2 dt$, where $i = 1, \cdots, 35$ and $\hat{\beta}_{-i}$ was estimated by the dataset without the ith observation. For the convenience of calculation, I computed $\|Y_i(t) - \eta_{\hat{\beta}_{-i}}(X_i, t)\|_2^2$ at a grid of values t as the surrogate of ISE. Since the findings through the coefficient function estimates were similar to those in Ramsay and Silverman (2005), I only focused on the comparison of prediction performance. The

Table 4.1: The mean, standard deviation and three quartiles of ISEs for the three approaches. The best result on each metric is in boldface.

Method	Median	Mean	Standard Deviation	1st Qu.	3rd Qu.
OPFFR	**21.6400**	**40.2800**	**45.7631**	**13.8000**	**36.1700**
FDA	25.9000	44.1600	56.9544	18.7400	40.6100
FPCA	30.7752	45.5065	45.7763	20.5031	52.1827

summary in Table 4.1 clearly showed the numerical advantage of the proposed OPFFR method over the FDA and FPCA methods.

Histone Regulation Data

Nucleosomes, the basic units of DNA packaging in eukaryotic cells, consist of eight histone protein cores including two copies of H2A, H2B, H3, and H4. Besides the role as DNA scaffold, histones provide a complex regulatory platform for regulating gene activity (Wollmann et al., 2012). Focused study of the interaction between histones and gene activity may reveal how the organisms respond to the environmental changes. There are multiple sequence variants of histone proteins, which have some amino acid changes compared to their primary sequence, coexist in the same nucleus. For instance, in both plants and animals, there exist three variants of H3, the H3.1, the H3.3, and the centromere-specific CENP-A (CENH3) (Deal and Henikoff, 2011). Each variant shows distinct regulatory mechanisms over gene expression.

In this thesis, an ultra-high throughput time course study was conducted to explore the interaction mechanism between the gene activity and histone variant, H3.3, during heat stress in *Arabidopsis thaliana*. In this study, the 12-day-old *Arabidopsis* seedlings that had been grown at 22 °C were subject to heat stress of 38 °C, and plants were harvested at 7 different time points within 24 hours for RNA sequencing (RNA-seq) (Wang et al., 2009) and ChIP sequencing (ChIP-seq) (Johnson et al., 2007) experiments. We were interested in the genes responding to the heat shock, therefore 160 genes in response to heat (GO:0006951) pathway (Ashburner et al., 2000) were chosen. I selected 55 genes with the fold change above 0.5 at at least two consecutive time points in RNA-seq data. In ChIP-seq experiments, I calculated the mean of normalized read counts by taking the average of normalized read counts over seven time points for the region of 350 base pairs (bp) in the downstream of transcription start sites (TSS) of selected 55 genes. The normalized read counts over 350 bp from ChIP-seq and the normalized fragments per kilobase of transcript per million mapped

reads (FPKM) (Trapnell et al., 2010) over seven time points from RNA-seq were used to measure the histone levels and gene expression levels respectively.

I applied the OPFFR, FDA and FPCA methods to histone regulation data in example 1.2.2. Since the gene expression levels were sparsely observed, I also applied the OPFFR-S method to the data. The comparison of the four methods is shown in Table 4.2. In the table, the standard deviation of ISEs was the only measure that neither the OPFFR nor the OPFFR-S was the most optimal. This was caused by a few observations where all the methods failed to make a good prediction and the OPFFR methods happened to have larger ISEs. In terms of all the other measures, the proposed OPFFR and OPFFR-S methods clearly showed the advantage in prediction accuracy again. Since the results from the OPFFR and OPFFR-S were comparable to each other, I chose to present all the following results based on the OPFFR analysis.

Table 4.2: The mean, standard deviation and three quartiles of ISEs for the four approaches. The best result on each metric is in boldface.

Method	Median	Mean	Standard Deviation	1st Qu.	3rd Qu.
OPFFR	1.5700	**7.7120**	18.9180	**0.5077**	**5.1900**
OPFFR-S	**1.4070**	7.7150	18.6037	0.6972	5.5820
FDA	2.2060	7.9770	18.7004	0.5461	6.2750
FPCA	2.0170	8.4720	**18.3978**	0.9126	6.1790

Figure 4.5 is the plot of the fitted coefficient function generated from our OPFFR method. For region between 300 bp and 350 bp, there was a strong negative influence of H3.3 on genes activity from half hour to 8 hours. It indicted that the loss of H3.3 might have the biological influence on the up-regulation of heat-induced genes. This negative correlation phenomenon was also observed after 30 minutes on the region of 250 bp to 300 bp between H3.3 and gene activity. In addition, the region from 50 bp to 150 bp had a positive effect on genes activity over time domain from 0 hour to half hour and 4 hours to 8 hours. Therefore, I provided a numerical evidence that heat-shock-induced transcription of genes in response to heat stress

might be regulated via the epigenetic changes of H3.3, especially on the downstream region of TSS. The sample plots in Figure 4.6 showed a nice match of the predicted gene expression curves with the observed values.

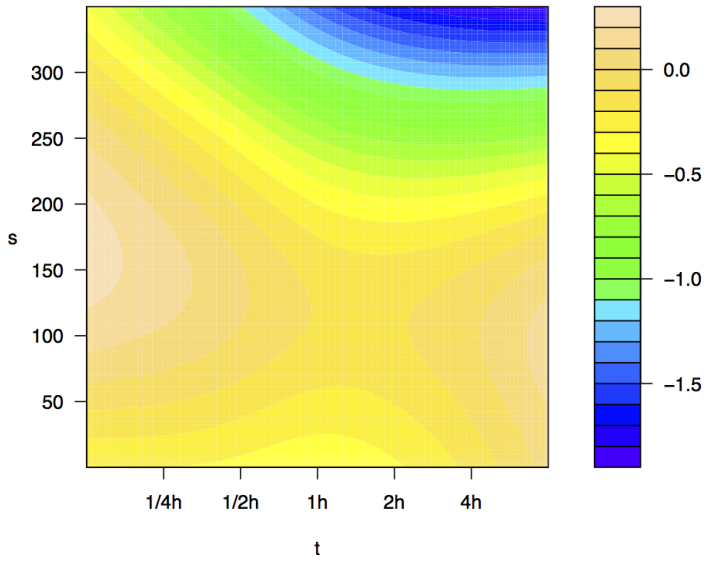

Figure 4.5: The estimated coefficient function $\beta(t, s)$ for the histone regulation study. The y-axis label represents the positions on genomes and x-axis label represents seven time points.

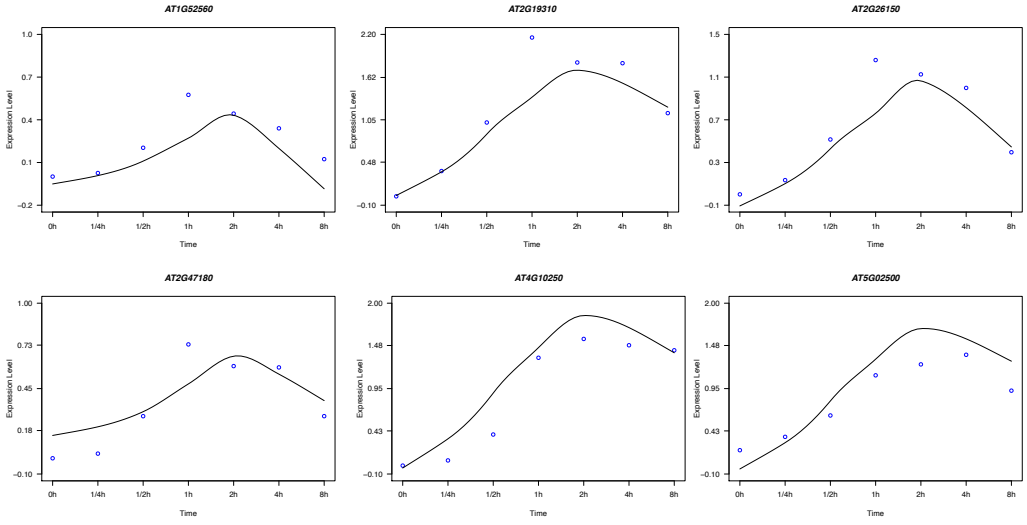

Figure 4.6: The fitted response functions for six genes in the histone regulation study. The y-axis stands for the normalized expression levels and x-axis label represents seven time points. The curve fitted using OPFFR is in the solid line, with the data in circles.

4.4 Discussion

In this thesis, I have presented a new analysis tool for modeling the relationship of a functional response against a functional predictor. The proposed method is more flexible and generally delivers a better numerical performance than the FPCA approach since it does not have the restrictive structural dependence assumption on the coefficient function. When compared with the penalized B-splines method, the proposed method has the theoretical advantage of possessing the optimal rate for mean prediction as well as some numerical advantage as shown in the numerical studies. Moreover, the Representer Theorem guarantees an exact solution to the penalized least squares, a property that is not shared by the existing penalized function-on-function regression models. The application of our method to a histone regulation study provided numerical evidence that the changes in H3.3 might regulate some genes through transcription regulations. Although such a finding sheds light on the relationship between histone variant H3.3 and gene activity, the details of the regulation process are still unknown and merit further investigations. For instance, we may investigate how the H3.3 organizes the chromatins to up-regulate those active genes. Such investigations would call for more collaborations between statisticians and biologists.

When the regression model has a scalar response against one or more functional predictors, methods other than the roughness penalty approach are available to overcome the inefficient basis representation drawback in the FPCA method. For example, Delaigle et al. (2012) considered a partial least squares (PLS) based approach. Ferré and Yao (2003) and Yao et al. (2015) translated the idea of sufficient dimension reduction (SDR) into the setting of functional regression models. Intuitively, these methods might be more efficient in their selection of the principal component basis functions since they incorporate the response information into consideration. However, the experiments with a functional response version of the functional PLS (Preda and Saporta, 2005), not shown here due to space limit, did not look so promising. Therefore, further investigation in this direction is surely needed.

Chapter 5

Statistical Inference for Time Course RNA-seq Data

Chapter Summary: Accurate identification of differentially expressed (DE) genes in time course RNA-seq data is crucial for understanding the dynamics of transcriptional regulatory network. However, most of the available methods treat gene expressions at different time points as replicates and test the significance of the mean expression difference between treatments or conditions irrespective of time. They thus fail to identify many DE genes with different profiles across time. In the thesis, I propose a negative binomial mixed-effect model (NBMM) under a RKHS framework to identify DE genes in time course RNA-seq data. In the NBMM, mean gene expression is characterized by a fixed effect, and time dependency is described by random effects. The NBMM is very flexible and can be fitted to both unreplicated and replicated time course RNA-seq data via a penalized likelihood method. By comparing gene expression profiles over time, I further classify the DE genes into two subtypes to enhance the understanding of expression dynamics. A significance test for detecting DE genes is derived using a Kullback-Leibler distance ratio. Additionally, a significance test for gene sets is developed using a gene set score. The materials of this chapter are mainly taken from Sun et al. (2016).

5.1 Negative Binomial Mixed-effect Model

In time course RNA-seq experiments, the short read counts cannot be adequately modeled by independent Gaussian distribution. I extend the aforementioned modeling strategy to develop a NBMM for modeling time course RNA-seq data.

The Model Specification

Suppose the time course RNA-seq experiments are conducted across G conditions/treatments. For each gene, the mapped read counts on exon k at time t_i in condition/treatment g, denoted by Y_{igk}, are assumed to follow a negative binomial distribution (NegBin),

$$Y_{igk} \sim \text{NegBin}(\nu, p(t_i, g, k)), \tag{5.1}$$

where the negative binomial distribution has the probability distribution,

$$P(Y_{igk} = y) = \frac{\Gamma(\nu + y)}{y!\Gamma(\nu)} p(t_i, g, k)^{\nu} (1 - p(t_i, g, k))^{y}, \tag{5.2}$$

where ν is a nuisance parameter, which is the number of reads that cannot be mapped to the reference genome, and $1 - p(t_i, g, k)$ is the probability that a read is mapped to exon k in condition g at time t_i, $g = 1, \cdots, G$, $i = 1, \cdots, n_g$, $k = 1, \cdots, \tilde{K}$. In this setting, n_g is the number of time points in the gth condition, and $\tilde{K}$ is the number of exons. In most cases, we only have two treatments: case and control or mutant and wild type ($G = 2$). To model the time trend and capture the time dependence, I use a nonparametric mixed-effect model with logit link (Gu, 2013, p.199)

$$\log\{p(t_i, g, k)/(1 - p(t_i, g, k))\} = \log(\beta_{t_i,g}) + \eta(t_i, g) + z_k b_k, \tag{5.3}$$

where $\beta_{t_i,g}$ is the effective library size, used in edgeR (Robinson and Oshlack, 2010), of the t_ith time point, mean expression η is assumed to be a smooth function of time t for each treatment g, z_k is the length of the kth exon, b_k represents the exon specific random effect to model the intra-exon variation with $b_k \sim N(0, \sigma^2)$, and the random effect variance σ^2 is to be estimated from the data. The $\log(\beta_{t_i,g})$ term provides a convenient device to normalize the reads to a common scale.

In model (5.3), the bivariate function η is decomposed as

$$\eta(t, g) = \eta_\emptyset + \eta_1(t) + \eta_2(g) + \eta_{1,2}(t, g), \tag{5.4}$$

where $\eta_\emptyset$ is the baseline expression irrespective of time and treatment, $\eta_1(t)$ is the time effect at time t, $\eta_2(g)$ is the treatment effect of the gth condition, and $\eta_{1,2}(t, g)$ is the interaction between time and treatment effects. The time and treatment effects are defined as the deviation from the baseline expression, and, therefore, $\int_0^T \eta_1(t)dt = 0$ and $\sum_{g=1}^G \eta_2(g) = 0$. Analogously, the time-treatment interaction is defined as $\int_0^T \eta_{1,2}(t, g)dt = 0$ for all g, and $\sum_{g=1}^G \eta_{1,2}(t, g) = 0$ for all t. This decomposition is referred to as the functional ANOVA decomposition described in Chapter 2. If the time-treatment interaction term $\eta_{1,2}(t, g)$ is significant, we have $\eta(t, g_1) - \eta(t, g_2) = \eta_2(g_1) - \eta_2(g_2) + \eta_{1,2}(t, g_1) - \eta_{1,2}(t, g_2)$ for every t. In the right hand side, the first two terms are constants and the remaining terms vary with t. When the time-treatment interaction $\eta_{1,2}(t, g)$ is not significant in (5.4), the model reduces to

$$\eta(t, g) = \eta_\emptyset + \eta_1(t) + \eta_2(g), \tag{5.5}$$

which produces the parallel population mean time course profiles for different treatment conditions, i.e., $\eta(t, g_1) - \eta(t, g_2) = \eta_2(g_1) - \eta_2(g_2)$ for each t, where the right hand side of the equation is a constant which does not vary with t. To distinguish the expression profiles, we define the genes with significant time-treatment interaction term in (5.4), i.e., $\eta_{1,2}(t, g) \neq 0$, as non-parallel differentially expressed (NPDE) genes. If genes have a significant main effect

in treatment g but no time-treatment interaction in (5.4), i.e., $\eta_2(g) \neq 0$ and $\eta_{1,2}(t,g) = 0$, we define those as parallel differentially expressed (PDE) genes (Ma et al., 2009).

Estimation

By (5.2), one has a minus log likelihood

$$\sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \{(\nu + Y_{igk}) \log(1 + e^{\log\{p(t_i,g,k)/(1-p(t_i,g,k))\}}) - \nu \log\{p(t_i,g,k)/(1-p(t_i,g,k))\}\}. \tag{5.6}$$

Substituting (5.3) into (5.6), we get the minus log likelihood of $\mathbf{Y}$ conditioning on random effects $\mathbf{b}$, where $\mathbf{Y} = (Y_{111}, \cdots, Y_{n_G,G,\tilde{K}})^t$, and $\mathbf{b} = (b_1, \cdots, b_{\tilde{K}})^t$. Therefore, the (Henderson) likelihood (Robinson, 1991) of $(\mathbf{Y}, \mathbf{b})$ is

$$\log(f_{y|b}(\mathbf{Y}|\mathbf{b}) f_b(\mathbf{b})) \propto \sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \{(\nu + Y_{igk}) \log(1 + e^{\log(\beta_{t_i}) + \eta(t_i,g) + z_k b_k}) - \nu[\log(\beta_{t_i}) + \eta(t_i,g) + z_k b_k]\} + \sum_{k=1}^{\tilde{K}} b_k^2/\sigma^2. \tag{5.7}$$

In (5.7), the $f_{y|b}$ denotes the conditional distribution (negative binomial) of $\mathbf{Y}$ given $\mathbf{b}$, and f_b denotes the distribution (normal) of $\mathbf{b}$. In the end, we derive a penalized (Henderson) likelihood (Gu and Ma, 2005, p.486) as

$$\sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \{(\nu + Y_{igk}) \log(1 + e^{\log(\beta_{t_i}) + \eta(t_i,g) + z_k b_k}) - \nu[\log(\beta_{t_i}) + \eta(t_i,g) + z_k b_k]\} + \sum_{k=1}^{\tilde{K}} b_k^2/\sigma^2 + \tilde{N}\lambda J(\eta), \tag{5.8}$$

where $\tilde{N} = \sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} n_g$, the quadratic functional $J(\eta)$ quantifies the smoothness of η, and the smoothing parameter λ controls the trade-off between the goodness-of-fit and the smoothness of η. The minimization of (5.8) is performed in a RKHS $\mathcal{H} \subseteq \{\eta : J(\eta) < \infty\}$, in which $J(\eta)$ is a square semi-norm, see details in Chapter 2. For model (5.3) with functional ANOVA (5.4), I employ the following quadratic penalty, which produces a cubic spline estimate,

$$J(\eta) = \theta_1^{-1} \int_0^T (d^2\eta_1/dt^2)^2 dt + \theta_{1,2}^{-1} \int_0^T \sum_{g=1}^{G} (d^2\eta_{1,2}/dt^2)^2 dt, \tag{5.9}$$

where θ_1 and $\theta_{1,2}$ are extra smoothing parameters that adjust the relative penalties on the roughness of different components. See detailed examples in Chapter 3. For model (5.3) with functional ANOVA (5.5), I use penalty

$$J(\eta) = \int_0^T (d^2\eta_1/dt^2)^2 dt. \tag{5.10}$$

To perform the penalized likelihood estimation of (5.8), I implement two nested iterative loops. Fixing the smoothing parameter, the inner loop minimizes (5.8), and the outer loop estimates the smoothing parameters and variance of random effects via the minimization of certain cross-validation score, see Gu and Ma (2005) for details. For fixed smoothing parameter λ, (5.8) can be minimized through Newton iteration. Write

$$l_{igk}(\breve{\zeta}_{igk}) = (\nu + Y_{igk}) \log(1 + e^{\breve{\zeta}_{igk}}) - \nu \breve{\zeta}_{igk}, \tag{5.11}$$

where $\breve{\zeta}_{igk} = \log(\beta_{t_i}) + \eta(t_i, g) + z_k b_k$. The quadratic approximation of $l_{igk}(\breve{\zeta}_{igk})$ at $\widetilde{\zeta}_{igk}$ is

$$\begin{aligned} l_{igk}(\breve{\zeta}_{igk}) &\approx l_{igk}(\widetilde{\zeta}_{igk}) + \widetilde{\mu}_{igk}(\breve{\zeta}_{igk} - \widetilde{\zeta}_{igk}) + \widetilde{\omega}_{igk}(\breve{\zeta}_{igk} - \widetilde{\zeta}_{igk})^2/2 \\ &= \widetilde{\omega}_{igk}(\widetilde{Y}_{igk} - \breve{\zeta}_{igk})^2/2 + E_{igk}, \end{aligned} \tag{5.12}$$

where $\widetilde{Y}_{igk} = \widetilde{\zeta}_{igk} - \widetilde{\mu}_{igk}/\widetilde{\omega}_{igk}$ and E_{igk} is independent of $\breve{\zeta}_{igk}$; $\widetilde{\mu}_{igk} = (\nu + Y_{igk})\widetilde{p}(t_i, g, k) - \nu$ and

$\widetilde{\omega}_{igk} = \nu(1 - \widetilde{p}(t_i, g, k))$. The Newton iteration can thus be performed via iterated weighted least squares,

$$\sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \widetilde{\omega}_{igk} (\widetilde{Y}_{igk} \ - \ \log(\beta_{t_i}) \ + \ \eta(t_i, g) \ + \ z_k b_k)^2 \ + \ \sum_{k=1}^{\tilde{K}} b_k^2/\sigma^2 \ + \ \tilde{N}\lambda J(\eta). \quad (5.13)$$

Since ν is unknown, we estimate it from data. We apply the log operation to (5.2), and drop the terms that do not involve ν to get the individual objective function. Then the joint objective function is the sum of minus individual objective functions,

$$\frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \{\log(\Gamma(\nu)) - \log \Gamma(\nu + Y_{igk}) - \nu \log(p(t_i, g, k))\}, \quad (5.14)$$

where Γ is the gamma function. Given $(Y_{igk}, p(t_i, g, k))$, one estimates ν via the minimization of (5.14). We iterate between the estimations of $\eta(x)$ and ν in (5.8) and (5.14) (Gu, 2013).

5.2 Statistical Inference

Single Gene Significance Testing

Once the model (5.3) is fitted to the exon level read counts data, we identify NPDE and PDE genes by testing the significance of the interaction and main effects in (5.4). To identify NPDE genes, we test the significance of the time-treatment interaction in (5.4), which is,

$$\mathrm{H}_0 : \ \eta_{1,2}(t, g) = 0; \quad \mathrm{H}_1 : \ \eta_{1,2}(t, g) \neq 0. \quad (5.15)$$

To derive the needed test statistic, we first define the Kullback-Leibler distance

$$KL(\eta, \hat{\eta}) \ = \ \frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{K}} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \{\frac{\nu}{p(t_i, g, k)} \log \frac{1 - p(t_i, g, k)}{1 - \hat{p}(t_i, g, k)} \ + \ \nu(\eta(t_i, g) \ - \ \hat{\eta}(t_i, g))\}. \quad (5.16)$$

Then, we use the following Kullback-Leibler distance ratio (KLR) (Gu, 2004) as our test statistic

$$KLR = \frac{KL(\hat{\eta}_F, \hat{\eta}_D)}{KL(\hat{\eta}_F, \eta_C)}, \tag{5.17}$$

where $\hat{\eta}_F$ stands for a full model estimate given that H_1 is true in the ANOVA decomposition (5.4), and $\hat{\eta}_D$ represents a reduced model estimate under the hypothesis that H_0 is true in (5.4). Analogously, we define η_C as a constant function. For genes that are not considered as NPDE by the preceding test, we further investigate whether they are PDE or not. In model (5.3) with functional ANOVA (5.5), we are interested in testing

$$H_0 : \ \eta_2(g) = 0; \quad H_1 : \ \eta_2(g) \neq 0. \tag{5.18}$$

In testing for PDE genes, the full model estimate $\hat{\eta}_F$ does not include a time-treatment interaction, and $\hat{\eta}_D$ only has an overall mean and time effect in (5.5).

The p values for identifying NPDE and PDE genes are calculated through a permutation procedure. First, we compute a Kullback-Leibler distance ratio KLR for a gene. Second, the time labels for the gene are shuffled, and we recompute the statistic for the shuffled gene. We repeat the second step $\tilde{M}$ times to obtain $KLR_1^*, \cdots KLR_{\tilde{M}}^*$. In the end, the p value for the gene is given by,

$$\#\Big\{KLR_i^* > KLR, i = 1, \cdots, \tilde{M}\Big\}/\tilde{M}, \tag{5.19}$$

where $\#\{\cdot\}$ represents the cardinality of the set, i.e., the number of permuted KLR^*s which is larger than the KLR.

Gene Set Significance Testing

In many studies, researchers are not only interested in identifying individual DE genes, but also in finding DE gene sets. A gene set may be defined by known biological information, for instance, a group of genes within the same biological pathway. Since genes within the same

gene set are closely related, we increase statistical power of significance tests by borrowing information across genes. In addition, we obtain more robust results from gene sets than from individual genes. Subramanian et al. (2005) proposed an approach named Gene Set Enrichment Analysis (GSEA), which tested the significance of pre-defined gene sets through a Kolmogorov-Smirnov like test. Efron and Tibshirani (2007) proposed gene set analysis (GSA), which was shown to make a significant improvement over GSEA.

Following the ideas from GSEA and GSA, we test for significant NPDE gene sets via the following steps. Initially, pre-defined gene sets $\mathcal{S}_1$, $\mathcal{S}_2$, ... , $\mathcal{S}_P$ are collected. Then, we compute the Kullback-Leibler distance ratio KLR based on (5.17) for all genes. For each gene set, $\mathcal{S}_k$, we calculate a gene set score, $\mathcal{G}_k$, defined as the average of the Kullback-Leibler distance ratios in (5.17),

$$\mathcal{G}_k = \sum_{i \in \mathcal{S}_k} KLR_i / \#\{\mathcal{S}_k\}, \tag{5.20}$$

where $\#\{\mathcal{S}_k\}$ is the number of genes in gene set $\mathcal{S}_k$. The gene set score $\mathcal{G}_k$ defines an enrichment test statistic, with a larger value of $\mathcal{G}_k$ suggesting a greater enrichment of NPDE genes. The PDE gene sets can be tested in the same way. To test the significance of the gene set, a threshold is needed. The following permutation procedure is used to determine the threshold, and gene sets with values of $\mathcal{G}_k$ above the threshold are declared significant. In particular, we shuffle the time label for each gene and recompute the statistic for each permuted gene. We utilize formula (5.20) to calculate the permuted gene set scores $\mathcal{G}_1^*, \cdots, \mathcal{G}_{\tilde{M}}^*$, where $\tilde{M}$ is permutation times. In the end, we calculate the p value of the kth gene set, given by,

$$\#\Big\{\mathcal{G}_i^* > \mathcal{G}_k, i = 1, \cdots, \tilde{M}\Big\}/\tilde{M}. \tag{5.21}$$

5.3 Numerical Experiments

Simulation Study

I evaluated the performance of the proposed method by carrying out extensive analysis on simulated datasets. Datasets were generated from both the NBMM model and an RNA-seq simulator. All p values were adjusted by Benjamini and Hochberg (BH) method for multiple testing corrections (Benjamini and Hochberg, 1995).

Single Gene Simulation

I simulated exon level read counts according to equation (5.1), (5.2) and (5.3). The effective library sizes of all time points were estimated by edgeR. We have three settings in this section. For each setting, $b_k \sim N(0,1), k = 1,2,3$, accounts for variation of different exons, $z_1 = 0.1$, $z_2 = 0.25$ and $z_3 = 0.4$ and ν is set to be 1000 for all those settings. Each exon was simulated with both single replicate and three replicates.

First setting: linear pattern. In the first setting, I generated exon level read counts of DE genes, see the top panel in the Figure 5.1, using the following function,

$$\eta(t_i, g) = C((0.9 - 2t_i)I_{[g=2]} + t_i), \tag{5.22}$$

where $t_i = i/10, i = 1, 2 \cdots, 8$, $g = 1, 2$, and $C = 2$ is a scale factor, $I_{[g=2]}$ is an indicator function which equals one when $g = 2$ and zero otherwise.

Second setting: exponential pattern. In the second setting, I simulated exon level read counts of DE genes, see the middle panel in the Figure 5.1, using the following smooth function,

$$\eta(t_i, g) = \exp\{10^4 F_1^{11} F_2^6 + 10^2 F_1^3 F_2^9 + C_g\}, \tag{5.23}$$

where $F_1 = (0.9 - 2t_i)I_{[g=2]} + t_i$, $F_2 = 0.1I_{[g=2]} + I_{[g=1]} + (1 - 2I_{[g=1]})t_i$, and $C_1 = C_2 = 1$. The constants C_g, $g = 1, 2$, define fixed reference expression levels for different conditions.

Third setting: cyclic pattern. In the third setting, exon level read counts of DE genes, see the bottom panel in Figure 5.1, were generated using the following smooth function,

$$\eta(t_i, g) = \sin(2.5\pi((0.9 - 2t_i)I_{[g=2]} + t_i) + 2. \tag{5.24}$$

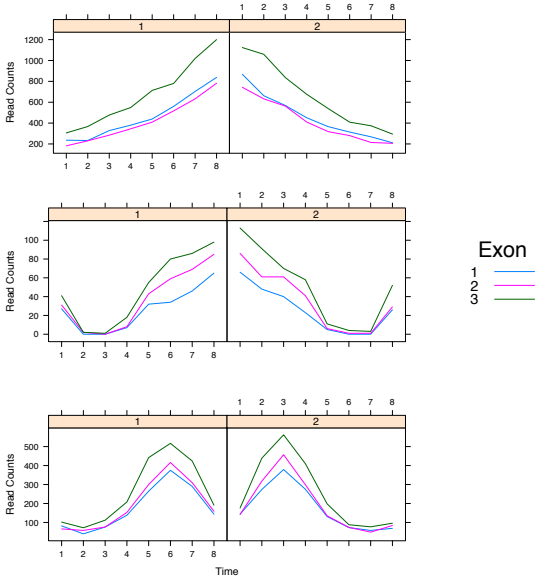

Figure 5.1: Simulated read counts generated from a negative binomial distribution. Samples of DE genes in the first, second and third setting are shown in the top, middle and bottom panels respectively. Different exons are represented by curves with varying colors.

There were two scenarios in each setting. In the first scenario, I simulated time course exon level read counts of 50 genes. Half of the genes were DE genes generated by the above mean functions, and the remaining genes were generated as non-differentially expressed (NDE) genes by using the same mean function for different conditions. In the second scenario, 25 DE genes had the same profiles as those in the first scenario and 225 NDE genes were modeled as flat profiles. I compared the NBMM with three methods, maSigPro (Nueda et al., 2014), DyNB (Äijö et al., 2014) and edgeR. The former two methods are designed for time course data. Analysis followed the steps described in the R package documentation and unless stated otherwise default parameters were used.

Table 5.1: The FDR and FNR of all methods for detecting DE genes in simulation studies. If the method failed to report any significant genes, the FDR was NA and FNR was 0.50 for scenario 1 and 0.09 for scenario 2.

			Setting 1		Setting 2		Setting 3	
			FDR	FNR	FDR	FNR	FDR	FNR
NBMM	Scenario 1	1 Rep	**0.00**	**0.00**	0.00	0.17	**0.00**	**0.14**
		3 Rep	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.14**
	Scenario 2	1 Rep	0.07	0.00	0.00	0.02	**0.21**	**0.02**
		3 Rep	0.16	0.00	0.15	0.01	**0.09**	**0.02**
maSigPro	Scenario 1	1 Rep	0.11	0.00	**0.00**	**0.00**	NA	0.50
		3 Rep	0.00	0.07	0.00	0.04	NA	0.50
	Scenario 2	1 Rep	**0.00**	**0.00**	**0.00**	**0.00**	NA	0.09
		3 Rep	0.00	0.01	0.00	0.01	NA	0.09
DyNB	Scenario 1	1 Rep	NA	0.50	0.00	0.36	NA	0.50
		3 Rep	0.54	0.54	0.32	0.32	0.43	0.20
	Scenario 2	1 Rep			NA			
		3 Rep			NA			
edgeR	Scenario 1	Rep 1			NA			
		3 Rep	0.50	NA	0.50	NA	0.50	NA
	Scenario 2	1 Rep			NA			
		3 Rep	0.88	0.00	**0.00**	**0.00**	0.86	0.00

Table 5.2: The running CPU time (seconds) for all methods in simulation studies.

		Setting 1	Setting 2	Setting 3
NBMM	1 Rep	7.133	6.182	7.261
	3 Rep	6.240	6.271	7.000
maSigPro	1 Rep	0.215	0.025	0.200
	3 Rep	0.235	0.091	0.236
DyNB	1 Rep	31944.470	NA	42513.210
	3 Rep	36228.200	36335.970	40412.250
edgeR	1 Rep	0.004	0.001	0.001
	3 Rep	0.001	0.001	0.001

Table 5.1 summarizes the performance of each method. The FDR was calculated as the number of false positives divided by the number of identified DE genes, and the False Non-Discovery Rate (FNR) as the number of false negatives divided by the number of genes which were not identified as DE genes. DyNB was only applied to the simulated dataset of the first scenario in each setting due to its extensive computational cost, see Table 5.2. In the third setting, the DyNB failed to report the results for the dataset with one replicate. In addition, edgeR was not recommended for single replicate datasets and, therefore, not used in each single replicate dataset (Nueda et al., 2014).

The performance of edgeR, DyNB and maSigPro in terms of FDR and FNR was not as good as that of NBMM in the first scenario. This is expected since edgeR is not designed for time course data and the accuracy of detecting DE genes is affected by the estimated effective library size. When the NDE genes do not show flat profiles, the prediction performance of edgeR and maSigPro relying on TMM normalization (Robinson and Oshlack, 2010) will be impaired. maSigPro had a better performance compared with NBMM method in the second scenario in linear and exponential settings. However, the proposed method performed much better than other methods in more complicated patterns, such as a cyclic pattern. For this pattern, other methods either failed to detect any DE genes or identified almost all the genes as DE genes. In particular, in the first setting, the proposed NBMM method identified all DE genes. In the third setting, the proposed approach identified about 88% of DE genes with FDR 0.00 in the first scenario, whereas the maSigPro failed to detect any DE genes. In summary, as the pattern of the mean function moves away from linear to nonlinear, the advantage of the NBMM over other methods is getting more significant in detecting DE genes. The NBMM took 7 seconds (CPU time) to process 50 genes with three replicates. Running CPU time for other settings are shown in Table 5.2. In summary, edgeR is not designed for time course RNA-seq data, and, therefore, their performance is not as good as that of the NBMM and maSigPro in most settings. The maSigPro is applicable to time course RNA-seq data and has a good performance in the roughly linear pattern. Its performance

in the highly nonlinear pattern is not as good as the NBMM.

Simulation using RNA-seq Simulator

An RNA-seq simulator, polyester (Frazee et al., 2015), was applied to simulate RNA-seq experiments. The simulator takes a set of annotated transcripts as input and produces files containing simulated RNA-seq reads after simulating the steps of an RNA-seq experiment. The reference genome used in the simulation was from *Drosophila melanogaster*. Tophat (Trapnell et al., 2012), samtools (Li et al., 2009) and DEXSeq (Anders et al., 2012) were utilized to estimate the read counts data from the simulated fasta files. Analysis followed the steps described in the documentations and unless stated otherwise, default parameters were used. I simulated the data of 7763 transcripts. By directly specifying the number of reads in each transcript, I simulated two expression patterns, linear expression pattern in (5.25) and nonlinear expression pattern in (5.26). In each pattern, 125 DE genes were created.

$$\upsilon_{t_i,g} = r((5 - t_i)I_{[g=2]} + t_i), \tag{5.25}$$

where r is the reference expression level defined in (5.27) and $t_i = 1 + 3(i-1)/7$.

$$\upsilon_{t_i,g} = r(\sin(2.5\pi((0.9 - 2t_i)I_{[g=2]} + t_i) + 2). \tag{5.26}$$

The reference expression level is

$$r = 20\iota/\upsilon, \tag{5.27}$$

where ι is the length of transcript and $\upsilon = 100$ is the length of short reads. The expression values for NDE genes in all time points are defined in (5.27). Removing genes with zero expression values over all time points, we came down with a dataset including 4526 genes, among which 219 genes were DE genes. I applied NBMM, maSigPro and edgeR to the dataset and results were summarized in Table 5.3. NBMM and maSigPro detected all DE

genes with linear change pattern, however, NBMM identified 40 DE genes with nonlinear pattern whereas maSigPro found no genes with this pattern. As we can see in Table 5.3, the FDR and FNR of NBMM are lower than those of maSigPro. edgeR identified almost all the genes as DE genes and resulted in a higher FDR in Table 5.3.

Table 5.3: The FDR and FNR of all methods for detecting DE genes in simulation using polyester.

	FDR	FNR
NBMM	**0.621**	**0.018**
maSigPro	0.737	0.028
edgeR	0.925	0.00

Gene Sets Simulation

In this study, I simulated 30 gene sets, each with ten genes. All 100 genes in the first ten gene sets were NPDE genes generated by the first setting in (5.22). The rest of the gene sets were NDE genes with the same mean function for two conditions. I chose $\nu = 1000, C = 2$ and calculated the gene set scores and p values for the simulated data. The R package GSA developed in Efron and Tibshirani (2007) was used to detect DE genes enriched gene sets.

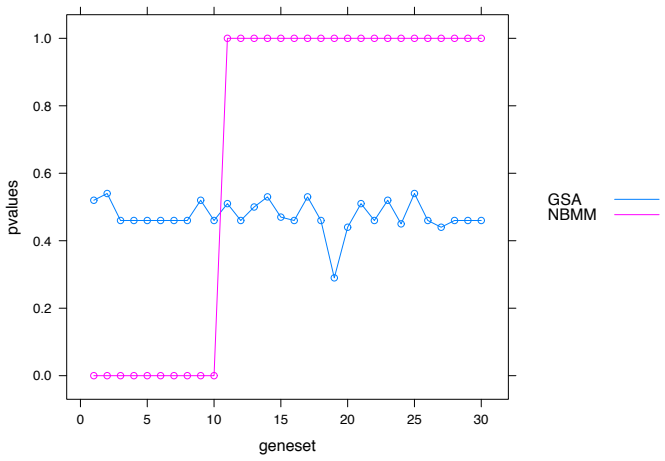

Figure 5.2: The p values of the proposed method are shown as pink cycles. The p values from GSA are shown as blue circles. The x-axis represents the gene set index, and the first 10 gene sets are the NPDE gene enriched gene sets.

In GSA package, I set *method*="mean", *minsize*=10, *resp.type*="two class unpaired" and other parameters as default. The p values for all 30 gene sets calculated by NBMM and GSA are plotted in Figure 5.2. The NBMM method detected all NPDE genes enriched gene sets, whereas the GSA method did not identify any significant gene sets.

Real Data Analysis

Study of the development of *Drosophila melanogaster* (fruit fly) is important since this biological process shares many common features among different organisms. Graveley et al. (2011) reported a time course RNA-seq experiment of *Drosophila melanogaster* embryogenesis. The dataset included 12 embryonic samples collected at 2-hour intervals for 24 hours. Each sample was collected at different stages of development. Sequencing was performed using the Illumina Genome Analyzer II platform. Reads of length 75 were uniquely aligned to the *Drosophila melanogaster* r5 genome using Bowtie (Langmead et al., 2009). Since in the first six time points, fruit flies were in the cleavage and gastrulation processes, whereas in the remaining six time points, they were in the process of differentiation (Campos-Ortega and Hartenstein, 1997), I divided the 12 time points into two developmental stages: early and late embryonic developmental stages. After data screening, the dataset used in our analysis consists of 1900 genes with different numbers of exons. Among these 1900 genes, 161 genes are related to embryo development (GO: 0009790) (Ashburner et al., 2000). I aim to identify DE genes between the two developmental stages and find the significant pathways.

Single Gene Testing

The NBMM model was fitted gene-by-gene and the KLRs were calculated. The permutation procedure was used to obtain the p value for each individual gene. After multiple testing corrections, our method identified 192 NPDE genes and 751 PDE genes at a significance level of 0.05. I conducted functional annotation clustering for these genes using DAVID (Huang et al., 2009). For NPDE genes, eight annotation clusters with enrichment scores above 2.0

were found. Seven of them are related to embryo development. For PDE genes, ten annotation clusters with enrichment scores above 2.0 were found. These clusters are associated with the regulation of RNA splicing, mitosis, and development related pathways. Moreover, edgeR was applied to this dataset and 518 DE genes were found. There were 292 genes in common between the edgeR and proposed approach, see Figure 5.3. Therefore, 651 DE

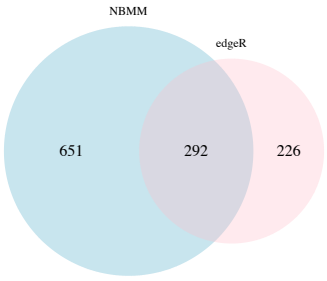

Figure 5.3: The Venn diagram between the sets of DE genes identified by NBMM and edgeR.

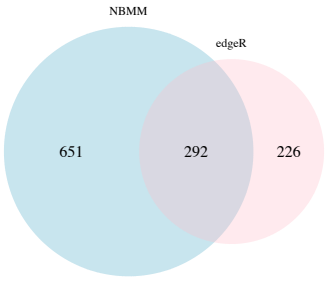

Figure 5.4: The Venn diagram between the sets of DE genes identified by NBMM and maSigPro.

genes were specifically found by NBMM and 226 DE genes were identified exclusively by edgeR. Among 161 genes in embryo development (GO: 0009790), 86 genes were identified by NBMM method, whereas edgeR detected 39 genes. For genes exclusively selected by edgeR, only two clusters with enrichment scores above 2.0 were found. These clusters are associated with certain catabolic processes. However, there were 11 clusters with enrichment scores above 2.0 for DE genes exclusively identified by the NBMM method. The biological processes associated with the clusters are the regulation of mRNA processing, mitosis,

nuclear division, determination of anterior/posterior axis, embryo, and neuroblast differentiation, etc. In addition, I compared the NBMM with maSigPro, which detected 1012 DE genes. There were 588 genes in common between these two models, see Figure 5.4. The NBMM specifically found 355 DE genes and 424 DE genes were identified exclusively by maSigPro. The annotation clustering was applied to these specifically identified DE genes. For genes exclusively selected by maSigPro, five clusters with enrichment scores above 2.0 were found. These clusters are associated with neuron projection morphogenesis , regulation of nuclear mRNA splicing and stem cell maintenance, etc. There were three clusters with enrichment scores above 2.0 for DE genes exclusively identified by the NBMM. The biological processes associated with the clusters are the mitosis, embryonic hindgut morphogenesis, gut development , etc.

Gene Sets Testing

The pathway gene sets of the fruit fly were compiled using the Bioconductor package "org.Dm.eg.db". The Entrez Gene identifier (version in Nov 2012) in each gene ontology term of org.Dm.egGO2ALLEGS was converted to official gene symbols using the org.Dm.egSYMBOL. I selected the gene sets with 15 to 30 genes and at least five of the 1900 genes were in the gene sets. I performed 100 permutations and chose pathways at the significance level of 0.05. Among 340 tested gene sets, 22 NPDE gene sets were selected by the NBMM, and 18 significant gene sets were selected by the GSA. Among 22 NPDE gene sets, eight gene sets are involved in the cell differentiation and cell development, see Table 5.4. The 18 significant gene sets detected by the GSA are the induction of apoptosis, chromosome localization, establishment of chromosome localization, cytoskeletal anchoring at plasma membrane, sarcomere organization, etc. These 18 gene sets are not associated with embryonic pathways. This shows that gene sets detected by the NBMM are more biologically relevant to development.

Table 5.4: The significant pathways identified by the NBMM gene set analysis of the fruit fly data.

Pathway Name	p value
segment polarity determination	0.00
salivary gland boundary specification	0.00
glial cell differentiation	0.00
glial cell development	0.00
axon choice point recognition	0.00
epithelial cell differentiation	0.00
regulation of tube length, open tracheal system	0.00
establishment of blood-brain barrier	0.00

5.4 Discussion

Time course RNA-seq data provide valuable insights into biological development and identifying biologically relevant DE genes is a key issue. We classify DE genes into two types: NPDE and PDE genes. Compared with PDE genes, NPDE genes are more likely to be biologically relevant. Therefore, focused study of the NPDE genes may provide more information on the underlying biological mechanisms. In this thesis, I proposed a statistical method, NBMM, for identifying DE genes in time course RNA-seq experiments. Compared to other available methods, such as edgeR, the NBMM models time dependency and exon variation using a mixed-effect model. Moreover, the proposed NBMM method outperforms other approaches designed for time course RNA-seq data in terms of DE genes detection accuracy, such as maSigPro and DyNB. The advantage of the NBMM over other competing methods is significant when they are applied to single replicate time course RNA-seq data. Furthermore, gene sets significance test is shown to effectively detect DE gene sets. The NBMM method is applied to gene expression data on a gene-by-gene basis. Thus, parallel computing can be employed for testing the significance of multiple genes simultaneously.

Chapter 6

Derivations and Proofs

6.1 Derivations of Smoothing Matrix $A(\lambda)$

Suppose the subsample size is b, the functional (3.1) is minimized for

$$(K + b\lambda I)\mathbf{c} + T\mathbf{d} = \mathbf{Y}$$
$$T^t\mathbf{c} = 0.$$

Let

$$T = \begin{pmatrix} F_1 & F_2 \end{pmatrix} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$$

be the QR-decomposition of T where F_1 and F_2 are orthogonal matrices and $\tilde{R}$ is an upper-triangular matrix. It is trivial to show that the symmetric smoothing matrix $A(\lambda)$ which satisfies

$$\begin{pmatrix} \eta_{b,\lambda}(x_1) \\ \vdots \\ \eta_{b,\lambda}(x_b) \end{pmatrix} = A(\lambda)\mathbf{Y},$$

has the representation

$$A(\lambda) = I - b\lambda F_2(F_2^t K F_2 + b\lambda I)^{-1} F_2^t.$$

Let the eigendecomposition of $F_2^t K F_2$ be UDU^t. Some algebra yields

$$\begin{aligned} I - A(\lambda) &= b\lambda F_2^t (F_2 K F_2^t + b\lambda U U^t)^{-1} F_2 \\ &= b\lambda F_2^t (UDU^t + b\lambda U U^t)^{-1} F_2 \\ &= b\lambda F_2^t U^t (D + b\lambda I)^{-1} U F_2 \\ &= b\lambda Z (D + b\lambda I)^{-1} Z^t, \end{aligned}$$

where $Z_{b\times b-M} = F_2^t U^t$ satisfies $Z'Z = I_{b-M}$, D_{b-M} is a diagonal matrix with entries $\zeta_{\nu b}$. Such a representation can be found in both special and generalized smoothing spline models.

6.2 Proof of Theorem 3.3.1

Define

$$\begin{pmatrix} h_{1,b} \\ \vdots \\ h_{b-M,b} \end{pmatrix} = Z^t \begin{pmatrix} \eta(x_1) \\ \vdots \\ \eta(x_b) \end{pmatrix},$$

as $b \to \infty$, and $\lambda \to 0$ for some $r > 1$, we have

$$\begin{aligned} \frac{1}{b} \operatorname{Tr} A^j(\lambda) &\doteq \frac{1}{b} \sum_{\nu=1}^{b-M} \left(\frac{\zeta_{\nu b}}{b\lambda + \zeta_{\nu b}}\right)^j \\ &= \frac{\tilde{l}_j}{b\lambda^{1/r}}(1 + o(1)), \quad \text{for} \quad j = 1, 2, \end{aligned}$$

where $\tilde{l}_j$ are some constants independent of b. Let $\mu_1(\lambda) = \frac{1}{b}\sum_{\nu=1}^{b-M} \frac{\zeta_{\nu b}}{b\lambda+\zeta_{\nu b}}$, $\mu_2(\lambda) = \frac{1}{b}\sum_{\nu=1}^{b-M}(\frac{\zeta_{\nu b}}{b\lambda+\zeta_{\nu b}})^2$, and $q(\lambda) = \frac{1}{b}\sum_{\nu=1}^{b-M}(\frac{b\lambda}{b\lambda+\zeta_{\nu b}})^2 h_{\nu b}^2$. The expectation of loss function in (3.4) can be written as

$$\begin{aligned}\mathbb{E}L(\lambda) &= \frac{1}{b}\left\|I - A(\lambda)\eta\right\|^2 + \frac{\sigma^2}{b}\operatorname{Tr} A^2(\lambda) \\ &\doteq \frac{1}{b}\sum_{\nu=1}^{b-M}(\frac{b\lambda}{b\lambda+\zeta_{\nu b}})^2 h_{\nu b}^2 + \sigma^2\mu_2(\lambda) \\ &\doteq q(\lambda) + \sigma^2\mu_2(\lambda). \end{aligned} \tag{6.1}$$

Taking the first derivative of the right part of (6.1) and setting it to 0, we have

$$q'(\lambda) = \frac{\sigma^2\tilde{l}_2}{br}\lambda^{-(r+1)/r}\big(1+o(1)\big). \tag{6.2}$$

Next, the expectation of generalized cross-validation score can be written as

$$\begin{aligned}\mathbb{E}G(\lambda) &= \frac{\frac{1}{b}\left\|I - A(\lambda)\eta\right\|^2 + \sigma^2\big(1 - 2\frac{1}{b}\operatorname{Tr} A(\lambda) + \frac{1}{b}\operatorname{Tr} A^2(\lambda)\big)}{\big(1 - \frac{1}{b}\operatorname{Tr} A(\lambda)\big)^2} \\ &\doteq \frac{q(\lambda) + \sigma^2\big(1 - 2\mu_1(\lambda) + \mu_2(\lambda)\big)}{\big(1-\mu_1(\lambda)\big)^2}. \end{aligned} \tag{6.3}$$

Taking the first derivative of the right part of (6.3) and setting it to 0, we have

$$\begin{aligned}&\big[q'(\lambda) + \sigma^2\big(-2\mu_1'(\lambda) + \mu_2'(\lambda)\big)\big]\big(1-\mu_1(\lambda)\big)^2 \\ &+2\big(1-\mu_1(\lambda)\big)\mu_1'(\lambda)\big[q(\lambda) + \sigma^2\big(1 - 2\mu_1(\lambda) + \mu_2(\lambda)\big)\big] = 0,\end{aligned}$$

which is equivalent to

$$q'(\lambda) = -\sigma^2\mu_2'(\lambda)\left\{1 + \frac{2\mu_1'(\lambda)}{\mu_2'(\lambda)\big(1-\mu_1(\lambda)\big)}\left[\frac{q(\lambda)}{\sigma^2} + \mu_2(\lambda) - \mu_1(\lambda)\right]\right\},$$

or

$$q'(\lambda) = -\sigma^2\mu_2'(\lambda)\{1 + M(\lambda)\},$$

where

$$M(\lambda) = \frac{2\mu_1'(\lambda)}{\mu_2'(\lambda)\big(1-\mu_1(\lambda)\big)}\left[\frac{q(\lambda)}{\sigma^2} + \mu_2(\lambda) - \mu_1(\lambda)\right].$$

Since we have

$$\mu_1(\lambda) = \frac{\tilde{l}_1}{b\lambda^{1/r}}\big(1+o(1)\big), \qquad \mu_1'(\lambda) = \frac{-\tilde{l}_1}{br}\lambda^{-(r+1)/r}\big(1+o(1)\big),$$
$$\mu_2(\lambda) = \frac{\tilde{l}_2}{b\lambda^{1/r}}\big(1+o(1)\big), \qquad \mu_2'(\lambda) = \frac{-\tilde{l}_2}{br}\lambda^{-(r+1)/r}\big(1+o(1)\big),$$

and $q(\lambda)$ is of order $O(\lambda^p)$ under different "smoothness" conditions indexed by $p \in [1,2]$, it is easy to show that $M(\lambda) = O(1/b\lambda^{1/r}) + O(\lambda^p)$. As $\lambda \to 0$, $b\lambda^{1/r} \to \infty$,

$$\begin{aligned} q'(\lambda) &= -\sigma^2\mu_2'(\lambda)\{1+o(1)\} \\ &= \frac{\sigma^2\tilde{l}_2}{br}\lambda^{-(r+1)/r}\big(1+o(1)\big). \end{aligned} \tag{6.4}$$

It can be shown that the (6.2) and (6.4) have the same root as $\lambda \to 0$ and $b\lambda^{1/r} \to \infty$. Therefore, the $\breve{\lambda}$ and λ_g are of order $b^{-r/(pr+1)}$. The proof is completed by plugging the λ_g into the λ^*.

6.3 Proof of Theorem 3.3.2

We now state three commonly-used conditions for proving Theorem 3.3.2.

Condition 6.3.1 *The functional V is completely continuous with respect to J.*

The asymptotic convergence rates of penalized likelihood estimates minimizing (1.2) is usually characterized through an eigenvalue analysis of J with respect to V. This condition implies that there exist eigenvalues $\rho_\nu = \zeta_\nu^{-1} - 1 \to \infty$ and the associated eigenfunctions ψ_ν such that

$$V(\psi_\nu, \psi_\mu) = \delta_{\nu,\mu}, \quad J(\psi_\nu, \psi_\mu) = \rho_\nu\delta_{\nu,\mu},$$

where $\delta_{\nu,\mu}$ is the Kronecker delta (Weinberger, 1974; Silverman, 1982).

Condition 6.3.2 *For ν sufficiently large and some $\beta > 0$, the eigenvalues ρ_ν of J with respect to V satisfy $\rho_\nu > \beta\nu^r$, where $r > 1$.*

The growth rate condition for eigenvalues ρ_ν determines how fast λ should approach 0.

Condition 6.3.3 *$Var(\psi_\nu(X), \psi_\mu(X)) \leq C$ for some $C < \infty$, $\forall \nu, \mu$.*

This condition requires a uniform bound for the fourth moments of $\psi_\nu(X)$.

By Theorem 3.3.1, the proposed $\lambda^* = O(n^{-r/(pr+1)})$. Putting the two Theorems together, we conclude that under conditions 6.3.1, 6.3.2, 6.3.3 and $J(\eta_0) < \infty$ hold for some $p \in [1, 2]$ and $r > 1$, as $\lambda \to 0$ and $n\lambda^{2/r} \to \infty$, we have

$$(V + \lambda J)(\eta_{n,\lambda^*} - \eta_0) = O(n^{-pr/(pr+1)}).$$

6.4 Proof of Theorem 4.1.1

Proof. By the results in Cucker and Smale (2002), $R_{1y}Y_i \in \mathcal{H}_{1y}$ and $R_{1x}X_j \in \mathcal{H}_{1x}$ for all $i, j = 1, \ldots, n$. For any function $\beta \in \mathcal{H}$, we have the decomposition

$$\beta(t,s) = \Big\{ d^t_{\beta_y} \psi_y(t) + c^t_{\beta_y} (R_{1y}Y)(t) \Big\} \Big\{ d^t_{\beta_x} \psi_x(s) + c^t_{\beta_x} (R_{1x}X)(s) \Big\} + \rho(t,s), \tag{6.5}$$

where $\rho \in \mathcal{H} \ominus \big[\{\mathcal{H}_{0y} \oplus \text{span}(R_{1y}Y_i : i = 1, \ldots, n)\} \otimes \{\mathcal{H}_{0x} \oplus \text{span}(R_{1x}X_i : i = 1, \ldots, n)\} \big]$. By the orthogonality between the components, we have

$$\begin{aligned} J(\beta) &= \langle \beta, \beta \rangle_{\mathcal{H}_1} \\ &= \| d^t_{\beta_y} \psi_y(t) (R_{1x}X)^t(s) c_{\beta_x} \|^2_{\mathcal{H}_1} + \| c^t_{\beta_y} (R_{1y}Y)(t) \psi^t_x(s) d_{\beta_x} \|^2_{\mathcal{H}_1} \\ &\quad + \| c^t_{\beta_y} (R_{1y}Y)(t) (R_{1x}X)^t(s) c_{\beta_x} \|^2_{\mathcal{H}_1} + \| \rho \|^2_{\mathcal{H}_1} \end{aligned}$$

Since $X_i(s) \in L_2(I_x)$ for each $i = 1, \ldots, n$, we can decompose them as

$$X_i(s) = d_{ix}^t \psi_x(s) + x_{i,\mathcal{H}_{1x}}(s) + l_{ix}(s),$$

where $x_{i,\mathcal{H}_{1x}}$ is the projection of X_i into $\mathcal{H}_{1x}$ and $l_{ix} \in L_2(I_x) \ominus \mathcal{H}_x$.

By the reproducing property, $(R_{1x} x_{i,\mathcal{H}_{1x}})(s) = x_{i,\mathcal{H}_{1x}}(s)$. Hence $(R_{1x} X_i)(s) = x_{i,\mathcal{H}_{1x}}(s) + (R_{1x} l_{ix})(s)$. Then

$$\int_{I_x} x_{i,\mathcal{H}_{1x}}(s)\rho(t,s)ds = \int_{I_x} (R_{1x}X_i)(s)\rho(t,s)ds - \int_{I_x} l_{ix}(s)\{R_{1x}\rho(t,\cdot)\}(s)ds = 0,$$

since ρ and l_{ix} are orthogonal to their respective integrand companies. Also note that $\int_{I_x} \beta(t,s) l_{ix}(s) ds = \langle \beta(t,\cdot), l_{ix} \rangle_{L_2(I_x)} = 0$ since $\beta(t,\cdot) \in \mathcal{H}_x$ for any fixed t. Hence

$$\begin{aligned}
\int_{I_x} \beta(t,s) X_i(s) ds &= d_{ix}^t \int_{I_x} \psi_x(s)\beta(t,s)ds + \int_{I_x} x_{i,\mathcal{H}_{1x}}(s)\beta(t,s)ds \\
&= \Big\{ d_{\beta_y}^t \psi_y(t) + c_{\beta_y}^t (R_{1y}Y)(t) \Big\} \Big[d_{ix}^t \{ \int_{I_x} \psi_x(s)\psi_x^t(s)ds \} d_{\beta_x} \Big] \\
&\quad + \Big\{ d_{\beta_y}^t \psi_y(t) + c_{\beta_y}^t (R_{1y}Y)(t) \Big\} \int_{I_x} c_{\beta_x}^t (R_{1x}X)(s) x_{i,\mathcal{H}_{1x}}(s) ds,
\end{aligned}$$

which does not involve the function ρ. This shows that the objective functional (4.2) depends on ρ only through the penalty term $\lambda J(\beta)$. That is, ρ appears in (4.2) only as a separate additive part $\lambda \|\rho\|_{\mathcal{H}_1}^2$. Hence the minimizer $\hat{\beta}$ of (4.2) in $\mathcal{H}$ must have $\rho = 0$.

6.5 Proof of Theorem 4.2.1

Proof. Recall that $L_2(R^{1/2}) = \mathcal{H}(R)$. So there exist γ_0 and $\hat{\gamma}_\lambda$ such that $\gamma_0 = R^{1/2}(\beta)$ and $\hat{\beta}_{n\lambda} = R^{1/2}(\hat{\gamma}_\lambda)$. For brevity, we will assume that $\mathcal{H}(R)$ is dense in L_2, which ensures that γ_0 and $\hat{\gamma}_\lambda$ are uniquely defined. The proof of the general case proceeds in exactly the same fashion by restricting ourselves to $L_2/\ker(L_2(R^{1/2}))$.

Denote $C_n(s_1, s_2) = n^{-1} \sum_{i=1}^n X_i(s_1) X_i(s_2)$, $S_{yx}(t,s) = n^{-1} \sum_{i=1}^n Y_i(t) X_i(s)$, and $S_{\epsilon x}(t,s) =$

$n^{-1}\sum_{i=1}^{n}\epsilon_i(t)X_i(s)$. Let

$$\Pi_n\big((t_1,s_1),(t_2,s_2)\big)=\int\int\int R^{1/2}\big((t_1,s_1),(z,u)\big)\ C_n(u,v)\ R^{1/2}\big((t_2,s_2),(z,v)\big)dudvdz.$$

The optimal solution of (4.2) can be written as $\hat{\gamma}_\lambda=\Big(\lambda I+\Pi_n\Big)^{-1}R^{1/2}S_{yx}$. Observe that $R^{1/2}S_{yx}=\Pi_n\gamma_0+g_n$, where $g_n=R^{1/2}S_{\epsilon x}$. Hence, $\hat{\gamma}_\lambda=\Big(\lambda I+\Pi_n\Big)^{-1}\Big(\Pi_n\gamma_0+g_n\Big)$.

Define $\gamma_\lambda=\big(\lambda I+\Pi\big)^{-1}\Pi\gamma_0$. It follows from triangular inequality that

$$\|\hat{\gamma}_\lambda-\gamma_0\|_\Pi=\|\gamma_\lambda-\gamma_0\|_\Pi+\|\hat{\gamma}_\lambda-\gamma_\lambda\|_\Pi. \tag{6.6}$$

Let us first bound the first term in the right hand side of (6.6). Recall that the ϕ_k are the eigenfunctions of Π. Write $\gamma_0=\sum_{k=1}^{\infty}a_k\phi_k$. Then $\gamma_\lambda=\sum_{k=1}^{\infty}\frac{a_k\rho_k}{\lambda+\rho_k}\phi_k$, and

$$\|\gamma_\lambda-\gamma_0\|_\Pi^2=\sum_{k=1}^{\infty}\frac{\lambda^2a_k^2\rho_k}{(\lambda+\rho_k)^2}\le\lambda^2\max_{k\ge1}\frac{\rho_k}{(\lambda+\rho_k)^2}\sum_{k=1}^{\infty}a_k^2=O(\lambda)\|\gamma_0\|_{L_2}^2.$$

Next, let us bound the second term in the right hand side of (6.6). Recall that $(\lambda I+\Pi_n)\hat{\gamma}_\lambda=\Pi_n\gamma_0-g_n$. Observe that

$$\begin{aligned}
\gamma_\lambda-\hat{\gamma}_\lambda&=(\lambda I+\Pi)^{-1}(\lambda I+\Pi_n)(\gamma_\lambda-\hat{\gamma}_\lambda)+(\lambda I+\Pi)^{-1}(\Pi-\Pi_n)(\gamma_\lambda-\hat{\gamma}_\lambda)\\
&=(\lambda I+\Pi)^{-1}\Pi_n(\gamma_\lambda-\gamma_0)+\lambda(\lambda I+\Pi)^{-1}\Pi_n\gamma_\lambda+(\lambda I+\Pi)^{-1}g_n\\
&\qquad+(\Pi+\lambda I)^{-1}(\Pi-\Pi_n)(\gamma_\lambda-\hat{\gamma}_\lambda)\\
&=(\lambda I+\Pi)^{-1}\Pi_n(\gamma_\lambda-\gamma_0)+\lambda\Pi\gamma_0+(\lambda I+\Pi)^{-1}g_n\\
&\qquad+(\Pi+\lambda I)^{-1}(\Pi-\Pi_n)(\gamma_\lambda-\hat{\gamma}_\lambda)\\
&=(\lambda I+\Pi)^{-1}\Pi(\gamma_\lambda-\gamma_0)+(\lambda I+\Pi)^{-1}(\Pi_n-\Pi)(\gamma_\lambda-\gamma_0)+\lambda\Pi\gamma_0\\
&\qquad+(\lambda I+\Pi)^{-1}g_n+(\Pi+\lambda I)^{-1}(\Pi-\Pi_n)(\gamma_\lambda-\hat{\gamma}_\lambda).
\end{aligned}$$

Therefore,

$$\begin{aligned}\|\hat{\gamma}_\lambda - \gamma_0\|_\Pi \leq\ & \|(\lambda I + \Pi)^{-1}\Pi(\gamma_\lambda - \gamma_0)\|_\Pi + \|(\lambda I + \Pi)^{-1}(\Pi_n - \Pi)(\gamma_\lambda - \gamma_0)\|_\Pi \\ & + \lambda\|\Pi\gamma_0\|_\Pi + \|(\lambda I + \Pi)^{-1} g_n\|_\Pi \\ & + \|(\lambda I + \Pi)^{-1}(\Pi - \Pi_n)(\gamma_\lambda - \hat{\gamma}_\lambda)\|_\Pi.\end{aligned}$$

We need to bound these five terms on the right hand side individually. The following discussion will be similar to the proof of Theorem 2 in Cai and Yuan (2012). For example, applying Lemmas 1 and 2 in Cai and Yuan (2012),

$$\begin{aligned}\|(\lambda I + \Pi)^{-1}\Pi(\gamma_\lambda - \gamma_0)\|_\Pi &\leq \|\Pi^{1/2}(\lambda I + \Pi)^{-1}\Pi^{1/2}\|_{op}\|\Pi^{1/2}(\gamma_\lambda - \gamma_0)\|_{L_2} \\ &\leq \frac{1}{2}\lambda^{1/2}\|\gamma_0\|_{L_2},\end{aligned}$$

where $\|\cdot\|_{op}$ stands for the usual operator norm. By Lemmas 1 and 4 of Cai and Yuan (2012), we have

$$\begin{aligned}\|(\lambda I + \Pi)^{-1}(\Pi_n - \Pi)(\gamma_\lambda - \gamma_0)\|_\Pi &\leq \|\Pi^{1/2}(\lambda I + \Pi)^{-1}(\Pi_n - \Pi)\Pi^{-\nu}\|_{op}\|\Pi^{\nu}(\gamma_\lambda - \gamma_0)\|_{L_2} \\ &\leq O_p((n\lambda^{1/(2r)})^{-1/2}\lambda^\nu) = o_p((n\lambda^{1/(2r)})^{-1/2}),\end{aligned}$$

where $\nu > 0$ satisfying $2r(1-2\nu) > 1$. Similarly,

$$\begin{aligned}\|(\lambda I + \Pi)^{-1}(\Pi - \Pi_n)(\gamma_\lambda - \hat{\gamma}_\lambda)\|_\Pi &= o_p((n\lambda^{1/(2r)})^{-1/2}) \\ \|(\lambda I + \Pi)^{-1} g_n\|_\Pi &= O_p((n\lambda^{1/(2r)})^{-1/2}).\end{aligned}$$

Combining these facts with $\lambda\|\Pi\gamma_0\|_\Pi = O(\lambda)$ yields that

$$\left\|\gamma_\lambda - \hat{\gamma}_\lambda\right\|_\Pi = O_p\Big(n^{-2r/(2r+1)}\Big).$$

6.6 Proof of Theorem 4.2.2

Proof. Since any lower bound for a specific case yields immediately a lower bound for the general case, we only study the case when the $\epsilon_i(t)$ are Gaussian white noise process with mean zero and variance σ^2. Fix $\alpha \in (0, 1/8)$, it follows from Theorem 2.5 in Tsybakov (2009) that in order to establish the minimax lower bound for $\mathfrak{R}_n$ we need to check the following three conditions

(a). $\beta_{jn} \in \mathcal{H}(R)$, $j = 0, \ldots, M$,

(b). $\int_{I_y} \mathbb{E}^* \Big\{ \eta_{\beta_{jn}}(X,t) - \eta_{\beta_{kn}}(X,t) \Big\}^2 dt \geq 2s$, for $0 \leq j < k \leq M$,

(c). $\frac{1}{M} \sum_{j=1}^{M} \mathcal{K}(P_j, P_0) \leq \alpha \log M$, where P_j denotes the joint distribution of $\{(Y_i, X_i) : i = 1, \ldots n\}$ with $\beta_0 = \beta_{jn}$ and $\mathcal{K}(\cdot, \cdot)$ is the Kullback-Leibler distance between two probability measures.

We will specific M and s later. If (a), (b), and (c) are all satisfied, then the minimax lower bound for the rate of convergence of $\mathfrak{R}_n$ has the same order of s.

To verify part (a), let m be the smallest integer greater than $c_0 n^{1/(2r+1)}$ for a positive constant to be specified later. For a $\omega = (\omega_{m+1}, \ldots, \omega_{2m}) \in \{0,1\}^m$, let $\beta_\omega = \sum_{j=m+1}^{2m} \omega_j m^{-1/2} R^{1/2}(\phi_j)$. Since

$$\begin{aligned} \|\beta_\omega\|^2_{\mathcal{H}(R)} &= \| \sum_{j=m+1}^{2m} \omega_j m^{-1/2} R^{1/2}(\varphi_j) \|^2_{\mathcal{H}(R)} \\ &= \sum_{j=m+1}^{2m} \omega_j^2 m^{-1} \| R^{1/2}(\phi_j) \|^2_{\mathcal{H}(R)} \\ &\leq \sum_{j=m+1}^{2m} m^{-1} \| R^{1/2}(\phi_j) \|^2_{\mathcal{H}(R)} = 1, \end{aligned}$$

this shows that $\beta_\omega \in \mathcal{H}(R)$. The last equality is due to the fact that

$$\langle R^{1/2}(\phi_j), R^{1/2}(\phi_k) \rangle_{\mathcal{H}(R)} = \langle \phi_j, R(\phi_k) \rangle_{\mathcal{H}(R)} = \langle \phi_j, \phi_k \rangle_{L_2} = \delta_{jk},$$

where $\delta_{jk} = 1$ for $j = k$, and 0 for $j \neq k$. Further, the Varshamov-Gilbert bound shows that, for $m \geq 8$, there exists s subset $\Omega = \{\omega^0, \omega^1, \ldots, \omega^M\} \subseteq \{0,1\}^m$ such that $\omega^0 = \{0, \ldots, 0\}$,

$$d(\omega^j, \omega^k) \geq \frac{m}{8}, \quad \forall \ \ 0 \leq j < k \leq M,$$

where $d(\cdot, \cdot)$ is the Hamming distance between ω^j and ω^k, and $M \geq 2^{m/8}$.

Let us now verify part (b). For $\omega, \omega' \in \Omega$, direct calculation yields that

$$\begin{aligned}
&\int_{I_y} \mathbb{E}^* \Big\{ \eta_{\beta_\omega}(X,t) - \eta_{\beta_{\omega'}}(X,t) \Big\}^2 dt \\
&= \sum_{j=m+1}^{2m} \sum_{k=m+1}^{2m} m^{-1}(\omega_j - \omega'_j)(\omega_k - \omega'_k) \int\int\int \big\{ R^{1/2}(\phi_j)(t,s_1) R^{1/2}(\phi_k)(t,s_2) C(s_1,s_2) \big\} dt ds_1 ds_2 \\
&= \sum_{k=m+1}^{2m} m^{-1}(\omega_k - \omega'_k)^2 \rho_k \\
&\geq m^{-1} \rho_{2m} d(\omega, \omega') \geq \ c_1 m^{-1} (2m)^{-2r} m/8 \geq c_2 n^{-2r/(2r+1)}.
\end{aligned}$$

Hence s in part (b) is of order $n^{-2r/(2r+1)}$.

Finally, let us check part (c). observe that for any $\omega, \omega' \in \Omega$,

$$\begin{aligned}
\log(P_{\beta_{\omega'}} / P_{\beta_\omega}) = \frac{1}{\sigma^2} \sum_{i=1}^n \int \Big[\Big\{ Y_i(t) - \int \beta_\omega(t,s) X_i(s) ds \Big\} \int \{ \beta_\omega(t,s) - \beta_{\omega'}(t,s) \} X_i(s) ds \Big] dt \\
- \frac{1}{2\sigma^2} \sum_{i=1}^n \int \Big[\int \{ \beta_\omega(t,s) - \beta_{\omega'}(t,s) \} X_i(s) ds \Big]^2 dt.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathcal{K}(P_{\beta_{\omega'}}, P_{\beta_\omega}) &= \frac{n}{2\sigma^2}\mathbb{E}^* \int \Big[\int \{\beta_\omega(t,s) - \beta_{\omega'}(t,s)\} X_i(s) ds \Big]^2 dt \\
&= \frac{n}{2\sigma^2} \sum_{k=m+1}^{2m} m^{-1} (\omega_k - \omega_k')^2 \rho_k \\
&\leq \frac{n}{2\sigma^2} \rho_m \sum_{k=m+1}^{2m} m^{-1} (\omega_k - \omega_k')^2 \\
&\leq \frac{n}{2\sigma^2} m^{-2r} \leq c_3 n^{1/(2r+1)}.
\end{aligned}$$

This implies that $M^{-1} \sum_{j=1}^{M} \mathcal{K}(P_j, P_0) \leq c_3 n^{1/(2r+1)} \leq \alpha \log M$. This completes the proof of Theorem 4.2.2.

Bibliography

Äijö, T., V. Butty, Z. Chen, V. Salo, S. Tripathi, C. B. Burge, R. Lahesmaa, and H. Lähdesmäki (2014). Methods for time series analysis of RNA-seq data with application to human Th17 cell differentiation. *Bioinformatics 30*(12), i113–i120.

Anders, S. and W. Huber (2010). Differential expression analysis for sequence count data. *Genome Biol. 11*, R106.

Anders, S., A. Reyes, and W. Huber (2012). Detecting differential usage of exons from RNA-seq data. *Genome Res. 22*(10), 2008–2017.

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc. 68*(3), 337–404.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. (2000). Gene ontology: tool for the unification of biology. *Nat. Genet. 25*(1), 25–29.

Ausió, J. (2006). Histone variants–the structure behind the function. *Brief. Funct. Genomic. Proteomic. 5*(3), 228–243.

Aydin, D., M. Memmedli, and R. E. Omay (2013). Smoothing parameter selection for nonparametric regression using smoothing spline. *Eur. J. Pure Appl. Math. 6*(2), 222–238.

Benatia, D., M. Carrasco, and J.-P. Florens (2017). Functional linear regression with functional response. *J. Econom. 201*(2), 269–291.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 289–300.

Cai, T. T. and M. Yuan (2011). Optimal estimation of the mean function based on discretely sampled functional data: phase transition. *Ann. Stat. 39*(5), 2330–2355.

Cai, T. T. and M. Yuan (2012). Minimax and adaptive prediction for functional linear regression. *J. Amer. Statist. Assoc. 107*(499), 1201–1216.

Campos-Ortega, J. A. and V. Hartenstein (1997). *The Embryonic Development of Drosophila melanogaster*. Berlin: Springer.

Cox, D. D. (1984). Multivariate smoothing spline functions. *SIAM J. Numer. Anal. 21*(4), 789–813.

Cox, D. D. and F. O'Sullivan (1990). Asymptotic analysis of penalized likelihood and related estimators. *Ann. Stat. 18*(4), 1676–1695.

Crambes, C. and A. Mas (2013). Asymptotics of prediction in functional linear regression with functional outputs. *Bernoulli 19*(5B), 2627–2651.

Craven, P. and G. Wahba (1978). Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math. 31*(4), 377–403.

Cucker, F. and S. Smale (2002). On the mathematical foundations of learning. *Bull. Amer. Math. Soc. 39*(1), 1–49.

Cuevas, A., M. Febrero, and R. Fraiman (2002). Linear functional regression: the case of fixed design and functional response. *Canad. J. Stat. 30*(2), 285–300.

Deal, R. B. and S. Henikoff (2011). Histone variants and modifications in plant gene regulation. *Curr. Opin. Plant Biol. 14*(2), 116–122.

Delaigle, A., P. Hall, et al. (2012). Methodology and theory for partial least squares applied to functional data. *Ann. Stat. 40*(1), 322–352.

Du, P. and X. Wang (2014). Penalized likelihood functional regression. *Statist. Sin. 24*, 1017–1041.

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *In Constr. Theory Funct. Sever. Var.*, 85–100.

Efron, B. and R. Tibshirani (2007). On testing the significance of sets of genes. *Ann. Appl. Stat. 1*(1), 107–129.

Ferraty, F., A. Laksaci, A. Tadj, and P. Vieu (2011). Kernel regression with functional response. *Electron. J. Stat. 5*, 159–171.

Ferré, L. and A.-F. Yao (2003). Functional sliced inverse regression analysis. *Stat. 37*(6), 475–488.

Frazee, A. C., A. E. Jaffe, B. Langmead, and J. T. Leek (2015). Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics 31*(17), 2778–2784.

Graveley, B. R., A. N. Brooks, J. W. Carlson, M. O. Duff, J. M. Landolin, L. Yang, C. G. Artieri, M. J. van Baren, N. Boley, B. W. Booth, et al. (2011). The developmental transcriptome of Drosophila melanogaster. *Nature 471*(7339), 473–479.

Gu, C. (2004). Model diagnostics for smoothing spline ANOVA models. *Canad. J. Stat. 32*(4), 347–358.

Gu, C. (2013). *Smoothing Spline ANOVA Models (2nd Ed.)*. New York: Springer-Verlag.

Gu, C. and P. Ma (2005). Generalized nonparametric mixed-effect models: computation and smoothing parameter selection. *J. Comput. Graph. Stat. 14*(2), 485–504.

Gu, C. and C. Qiu (1993). Smoothing spline density estimation: theory. *Ann. Stat. 21*(1), 217–234.

Gu, C. and G. Wahba (1991). Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. *SIAM J. Sci. Statist. Comput. 12*(2), 383–398.

Hall, P. (1990). Using the bootstrap to estimate mean squared error and select smoothing parameter in nonparametric problems. *J. Multivariate Anal. 32*(2), 177–203.

Hastie, T. and R. Tibshirani (1986). Generalized additive models. *Statist. Sci. 1*(3), 297–310.

Helwig, N. E., Y. Gao, S. Wang, and P. Ma (2015). Analyzing spatiotemporal trends in social media data via smoothing spline analysis of variance. *Spat. Stat. 14*, 491–504.

Helwig, N. E. and P. Ma (2015). Fast and stable multiple smoothing parameter selection in smoothing spline analysis of variance models with large samples. *J. Comput. Graph. Stat. 24*(3), 715–732.

Huang, D. W., B. T. Sherman, and R. A. Lempicki (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res. 37*(1), 1–13.

Hurvich, C. M., J. S. Simonoff, and C.-L. Tsai (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *J. Roy. Statist. Soc. Ser. B 60*(2), 271–293.

Ivanescu, A. E., A.-M. Staicu, F. Scheipl, and S. Greven (2015). Penalized function-on-function regression. *Comput. Stat. 30*(2), 539–568.

Johnson, D. S., A. Mortazavi, R. M. Myers, and B. Wold (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science 316*(5830), 1497–1502.

Kim, Y.-J. and C. Gu (2004). Smoothing spline Gaussian regression: more scalable computation via efficient approximation. *J. Roy. Statist. Soc. Ser. B 66*(2), 337–356.

Kimeldorf, G. and G. Wahba (1971). Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl. 33*(1), 82–95.

Langmead, B., C. Trapnell, M. Pop, and S. Salzberg (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol. 10*(3), R25.

Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al. (2009). The sequence alignment/map format and samtools. *Bioinformatics 25*(16), 2078–2079.

Lian, H. (2015). Minimax prediction for functional linear regression with functional responses in reproducing kernel Hilbert spaces. *J. Multivariate Anal. 140*, 395–402.

Liang, X., T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen (2015). Assessing Beijing's $PM_{2.5}$ pollution: severity, weather impact, APEC and winter heating. *Proc. R. Soc. A 471*(2182).

Lin, Y. (2000). Tensor product space ANOVA models. *Ann. Stat. 28*(3), 734–755.

Ma, P., W. Zhong, and J. S. Liu (2009). Identifying differentially expressed genes in time course microarray data. *Stat. Biosci. 1*(2), 144–159.

Malfait, N. and J. O. Ramsay (2003). The historical functional linear model. *Canad. J. Stat. 31*(2), 115–128.

Mallows, C. L. (1973). Some comments on Cp. *Technometrics 15*(4), 661–675.

Meyer, M. J., B. A. Coull, F. Versace, P. Cinciripini, and J. S. Morris (2015). Bayesian function-on-function regression for multilevel functional data. *Biometrics 71*(3), 563–574.

Nason, G. P. (1996). Wavelet shrinkage using cross-validation. *J. Roy. Statist. Soc. Ser. B*, 463–479.

Nueda, M. J., S. Tarazona, and A. Conesa (2014). Next maSigPro: updating maSigPro bioconductor package for RNA-seq time series. *Bioinformatics 30*(18), 2598–2602.

Oh, S., S. Song, G. Grabowski, H. Zhao, and J. P. Noonan (2013). Time series expression analyses using RNA-seq: a statistical approach. *BioMed Res. Int. 2013*.

Pope III, C. A., R. T. Burnett, M. J. Thun, E. E. Calle, D. Krewski, K. Ito, and G. D. Thurston (2002). Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. *J. Amer. Med. Assoc. 287*(9), 1132–1141.

Preda, C. and G. Saporta (2005). PLS regression on a stochastic process. *Comput. Stat. Data Anal. 48*(1), 149–158.

Ramsay, J. O., G. Hooker, and S. Graves (2009). *Functional Data Analysis with R and MATLAB*. Springer Science & Business Media.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis*. New York: Springer-Verlag.

Rice, J. and M. Rosenblatt (1983). Smoothing splines: regression, derivatives and deconvolution. *Ann. Stat. 11*(1), 141–156.

Robinson, G. K. (1991). That BLUP is a good thing: The estimation of random effects. *Stat. Sci.*, 15–32.

Robinson, M. and A. Oshlack (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol. 11*(3), R25.

Robinson, M. D., D. J. McCarthy, and G. K. Smyth (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics 26*(1), 139–140.

Scheipl, F. and S. Greven (2016). Identifiability in penalized function-on-function regression models. *Electron. J. Stat. 10*, 495–526.

Silverman, B. W. (1982). On the estimation of a probability density function by the maximum penalized likelihood method. *Ann. Stat. 10*(3), 795–810.

Speckman, P. (1985). Spline smoothing and optimal rates of convergence in nonparametric regression models. *Ann. Stat. 13*(3), 970–983.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. 102*(43), 15545–15550.

Sun, X., D. Dalpiaz, D. Wu, J. S. Liu, W. Zhong, and P. Ma (2016). Statistical inference for time course RNA-seq data using a negative binomial mixed-effect model. *BMC Bioinform. 17*(1), 324.

Sun, X., P. Du, X. Wang, and P. Ma (2018). Optimal penalized function-on-function regression under a reproducing kernel hilbert space framework. *J. Amer. Statist. Assoc.*, in press.

Trapnell, C., A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat. Protoc. 7*(3), 562–578.

Trapnell, C., B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. Van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter (2010). Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotech. 28*(5), 511.

Wahba, G. (1975). Smoothing noisy data with spline functions. *Numer. Math. 24*(5), 383–393.

Wahba, G. (1977). Practical approximate solutions to linear operator equations when the data are noisy. *SIAM J. Numer. Anal. 14*(4), 651–667.

Wahba, G. (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Stat. 13*(4), 1378–1402.

Wahba, G. (1990). *Spline models for observational data*, Volume 59 of *CBMS-NSF Regional Conf. Ser. in Appl. Math.* Philadelphia: SIAM.

Wang, J.-L., J.-M. Chiou, and H.-G. Mller (2016). Functional data analysis. *Ann. Rev. Stat. and Its Appl. 3*(1), 257–295.

Wang, Y. (2011). *Smoothing splines: methods and applications*. CRC Press.

Wang, Z., M. Gerstein, and M. Snyder (2009). RNA-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet. 10*(1), 57–63.

Weinberger, H. F. (1974). *Variational methods for eigenvalue approximation*. SIAM.

Wollmann, H., S. Holec, K. Alden, N. D. Clarke, P.-E. Jacques, and F. Berger (2012). Dynamic deposition of histone variant H3.3 accompanies developmental remodeling of the Arabidopsis transcriptome. *PLoS Genet. 8*(5), e1002658.

Wood, S. N. (2000). Modelling and smoothing parameter estimation with multiple quadratic penalties. *J. Roy. Statist. Soc. Ser. B 62*(2), 413–428.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *J. Amer. Statist. Assoc. 99*(467), 673–686.

Yao, F., E. Lei, and Y. Wu (2015). Effective dimension reduction for sparse functional data. *Biometrika 102*(2), 421–437.

Yao, F., H.-G. Müller, and J.-L. Wang (2005a). Functional data analysis for sparse longitudinal data. *J. Amer. Statist. Assoc. 100*(470), 577–590.

Yao, F., H.-G. Müller, and J.-L. Wang (2005b). Functional linear regression analysis for longitudinal data. *Ann. Stat. 33*(6), 2873–2903.

Yuan, M. and T. T. Cai (2010). A reproducing kernel Hilbert space approach to functional linear regression. *Ann. Stat. 38*(6), 3412–3444.