ABSTRACT

AVERY, MATTHEW ROGERS. New Techniques for Functional Data Analysis: Model Selection, Classification, and Nonparametric Regression. (Under the direction of Hao Helen Zhang and Yichao Wu.)

Functional data, such as curves and images, are increasingly collected in many research disciplines due to rapid advances in computers and modern scientific technologies. Compared to traditional scalar- or vector-data, functional data are inherently infinite-dimensional and more complex, and the observations from random trajectories are often sparse and irregularly sampled. In such cases, traditional statistical methods can not be directly applied, and new methods must be developed. Here, we study three problems related the functional data.

First, we look at the problem of sparse regression for functional data. In traditional settings, sparse regression refers to model selection problems where a subset of a $p\times 1$ vector of predictor variables is chosen to model the response. In functional data, we consider a model with a scalar response, Y and associated functional predictor $X(t)$ observed over some domain $\mathcal{T}$. In this context, a sparse model is one in which the coefficient function $\beta(t)$ for $X(t)$ is set identically to 0 over some subset of $\mathcal{T}$. Rather than choose a subset of predictor variables, we choose a subset of the domain of a single functional predictor. We propose a method for sparse regression using the Fused LASSO with a 1st order b-spline basis. This two-stage method is flexible and can be used in conjunction with all methods for functional regression.

Next, we consider the problem of classification for functional data, which has important and broad applications in practice. A new discriminant analysis approach is developed and studied. The central idea is to first reduce data dimension by functional principle component analysis

(FPCA) and then conduct linear discriminant analysis (LDA) based on the FPCA scores. Compared with existing functional data classification methods, the new procedure is conceptually much simpler, easier and faster to implement, and competitive in performance. The demanded programming effort is minimal, as the procedure can take advantages of existing software packages. Theoretical justifications are provided for the procedure in terms of its classification consistency under certain conditions.

Finally, we consider nonlinear functional regression for sparse and irregular data. As in the traditional setting, most methods for functional regression impose specific model assumptions on the data. If the true relationship between predictor and response does not conform to these assumptions, the resulting estimates may be poor. We propose a nonlinear regression method for functional data. Unlike other nonparametric methods discussed in the literature, ours can be applied to sparse and irregularly sampled data because it uses PACE to estimate the predictor trajectories. We compare our method to existing parametric methods on simulated data as well as blood pressure data from the Baltimore Longitudinal Study of Aging.

New Techniques for Functional Data Analysis: Model Selection, Classification, and Nonparametric Regression

by
Matthew Rogers Avery

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

Statistics

Raleigh, North Carolina

2012

APPROVED BY:

Daowen Zhang	Ana-Maria Staicu
Hao Helen Zhang Co-chair of Advisory Committee	Yichao Wu Co-chair of Advisory Committee

BIOGRAPHY

Matthew Avery was born in Davis, CA to Joy and Michael Avery. He grew up in Gainesville, FL, graduating from Eastside High School with an IB Diploma in 2002. He received his Bachelor of Arts Degree in Economics from New College of Florida in 2006 after completing his dissertation, *The Welfare Effects of Direct to Consumer Advertising in Prescription Drug Markets*. He began studying statistics at NCSU in 2006, and now moves on to pursue a career at the Institute for Defense Analyses.

ACKNOWLEDGEMENTS

It has taken the efforts of many people for me to reach this point, and here I give an incomplete list. My family, Mom, Dad, and Lily, have constantly cheered for me and pushed me forward. I wouldn't be here without them. I got some great help as an undergraduate, particularly from my adviser, Dr. Richard Coe, as well as Dr. Catherine Elliott. I'm not sure anyone has done more to spur my interest in learning and academics than Dr. Elliott. There's no way I would've gotten through graduate school without the encouragement and support of my friends and classmates, Justin, Paul, Malorie, Laura, David, Adam, Frank and Carrie. Special thanks to Kat for her notes and suggestions. Finally, this dissertation would not have come together without my advisers, Dr. Helen Zhang and Dr. Yichao Wu, who've given me good direction at every turn.

Thank you to you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

As technology progresses and more and more data becomes available, statistical methods must keep up. In particular, new data types, such as images and functions, come with new challenges. Functional MRIs produce images that neuroscientists and psychologists would like to use to learn about psychological diseases and neural connectivity. Data that might previously have been treated as longitudinal is now being treated as functional data. In such situations, new methods must be devised that leverage the unique characteristics of the data.

For standard data, common problems include model selection, classification, and nonparametric regression. Model selection techniques are employed in a variety of cases, such as when one is unsure which predictors are (most) important for modeling the response. These methods choose a subset of predictors, resulting in a sparse model that is generally easier to interpret than the full model. We frame the model selection problem as choosing a subset of the domain of a functional predictor rather than a subset of different predictors. By identifying such a subset, we can limit the time points at which we must collect data to only the most relevant.

Also, interpreting the predictor's relationship with the response may be simpler, since it will be defined only over a certain range.

For standard data, k-class classification is well-studied, but this problem also arises when the predictors are curves. Particularly when the observed data are sparsely and irregularly sampled, simple methods for classification are desirable. Similarly, while linear and quadratic regression have been proposed for functional data, both methods rely on a specified parametric model. When the data does not conform to this model (as in the case of a complex nonlinear relationship), the resulting estimators may not be consistent. A more general approach, particularly in the case of sparse and irregularly sampled data, may be able to model such relationships more accurately.

1.1 Regression

The standard linear regression model for n observations and p predictors can be written as

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\boldsymbol{y}$ is a vector of n response variables, X is an $n \times p$ design matrix whose ith row, $\boldsymbol{x}_i^T = [x_{i1}, x_{i2}], \ldots, x_{ip}$, consists of the values of the p predictor variables for the ith observation, $\boldsymbol{\beta}$ is a vector of coefficients for the p predictors, and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of random errors. When necessary, we will assume X has been normalized and the responses centered. That is, $\sum_{i=1}^n y_i = 0$, $\sum_{i=1}^n x_{ij} = 0$, and $\sum_{i=1}^n x_{ij}^2 = 1$ for $j = 1, \ldots, p$.

From this general framework, many methods have been developed for estimating the coefficients, $\boldsymbol{\beta}$. The simplest method is to choose $\boldsymbol{\beta}$ to minimize the residual sum of squares. That is, minimize

$$||\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{x}_j \boldsymbol{\beta}_j||^2. \tag{1.2}$$

This is commonly known as Ordinary Least Squares (OLS) regression.

1.1.1 Penalization Methods

While OLS works well in simple cases, it may not always be appropriate. For example, when p is large, the model may be difficult to interpret as well as over-fit. A common approach in such cases is penalized regression, where rather than minimize 1.2, we minimize

$$||\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{x}_j \boldsymbol{\beta}_j||^2 + P_\lambda(\boldsymbol{\beta}), \tag{1.3}$$

where $P_\lambda(\boldsymbol{\beta})$ is a penalization function with parameter λ whose argument is the coefficient vector. Common examples of penalty functions include the Ridge Regression penalty (Hoerl and Kennard, 1970), which is the L^2 norm of the coefficient vector, and the LASSO (Least Absolute Shrinkage and Selection Operator) penalty (Tibshirani, 1994), which is the L^1 norm of the coefficient vector. Thus, to find the LASSO fit, we minimize

$$||\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{x}_j \boldsymbol{\beta}_j||^2 + \lambda||\boldsymbol{\beta}||. \tag{1.4}$$

These penalization methods yield coefficient estimates that have simpler interpretations than the standard OLS fit. For example, when a regression problem has many predictor variables (that is, p is large), many of the estimated coefficients may be small. Small marginal effects can be difficult to interpret, since while they may be "statistically significant", they may not be significant in a practical sense. The LASSO addresses this issue by setting these small marginal effects to be exactly zero. The resulting fit is simpler to interpret than the OLS fit. Indeed, the LASSO is performing variable selection, since relatively few predictors are "selected" to be in the estimated model. The resulting model is called "sparse", since many predictor variables have been weeded out, and we are left with a smaller "sparser" model.

Since the LASSO was first described, numerous methods have been devised using this penalized regression framework. The Adaptive LASSO (Huang et al., 2006) uses weights from, for example, the standard OLS fit to scale the penalty applied to each predictor coefficient. The Adaptive LASSO also enjoys the Oracle property, which the standard LASSO does not. The Elastic Net (Zou and Hastie, 2005) is useful for selecting groups of highly correlated variables. While the standard LASSO will typically select only one of these predictors and exclude the rest, the Elastic Net is designed to include the whole group in the estimated model.

Finally, the Fused LASSO (Tibshirani et al., 2005) finds sparse fits for models where consecutive predictors may be included in the model. This is done by penalizing the differences between consecutive coefficients in addition to penalizing the coefficients directly. For this approach, we must presuppose that the predictor variables conform to an underlying structure where successive variables are somehow related. That is, if the jth variable is important to

predicting the response, y, then we believe the $(j-1)$th and $(j+1)$th variables may also be important. This occurs when the predictor variables have some relevant ordering to them, such as the case when the predictors represent observations at consecutive time points. Alternatively, the ordering can be constructed by placing correlated predictors near one another. (Tibshirani et al., 2005) To fit the Fused LASSO, we minimize

$$||\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{x}_j \boldsymbol{\beta}_j||^2 + \lambda_1 ||\boldsymbol{\beta}|| + \lambda_2 ||\boldsymbol{\beta}_{diff}||, \tag{1.5}$$

where $\boldsymbol{\beta}_{diff} = (\beta_2 - \beta_1, \beta_3 - \beta_2, \ldots, \beta_p - \beta_{p-1})$ and λ_1, λ_2 are penalization parameters. The resulting fits are characterized by large ranges of consecutive coefficients set identically to 0. The Fused LASSO has applications for functional data which will be explored in Chapter 2.

1.1.2 Nonparametric Regression

The previous discussion has focused on the subset of regression models where a particular relationship between the predictors and the response is assumed prior to model fitting. While this has the advantage of producing good model estimates when the true relationship is consistent with the model assumptions, when these assumptions are violated, the resulting estimates may be biased. One solution to this is nonparametric regression. In this approach, the only assumption is that

$$y_i = \mu(\boldsymbol{x}_i) + \epsilon_i. \tag{1.6}$$

Once again, $\boldsymbol{x}_i$ is a vector of predictor variables for y_i, and ϵ_i is the random error for the ith observation. To estimate $\mu(\cdot)$, a common approach is kernel regression. Let $K(\cdot)$ be a suitable (non-negative, symmetric, with $\int K(t)dt = 1$) kernel and h_n be an associated bandwidth. Then a nonparametric estimator for y^* with associated predictor vector $\boldsymbol{x}^*$ is

$$\frac{\sum_{i=1}^{n} Y_i K(\frac{\boldsymbol{x}^* - \boldsymbol{x}_i}{h_n})}{\sum_{i=1}^{n} K(\frac{\boldsymbol{x}^* - \boldsymbol{x}_i}{h_n})}. \tag{1.7}$$

This is the multivariate version of the Nadaraja-Watson estimator. See also Fan and Gijbels (1996). The choice of bandwidth and kernel function are important, and have been discussed in Rice (1984) and Fan and Gijbels (1992).

1.2 Functional Data

In both genetics and biostatistics, multiple observations on a single subject over time are increasingly being treated as realizations from a single underlying process. This so-called functional data has motivated numerous new techniques that leverage its unique characteristics to fit more accurate models. A general overview of functional data analysis including some fundamental methods was given by Ramsay and Silverman (2005).

Generally, a functional variable can be defined as a smooth process that occurs over some domain, typically time. When modeling functional data, both response variables as well as predictors can be functional, though we focus here on the case of a scalar response with an associated functional predictor.

Ideally, we would want to observe the functional variable over its entire domain. This is the case with Canadian weather data discussed in Ramsay and Silverman (2005). For each of 35 regions, average daily temperatures are recorded, giving us a temperature trajectory over the course of a full year. In this case, the trajectories are said to be regularly observed on a dense grid of time points. These curves are given in Figure 1.1a.

The Canadian weather data is an ideal situation where the grid of observed points is sufficiently dense that the underlying process can easily be interpolated. It is more common to observe fewer points for each trajectory. For example, Spellman et al. (1998) look at temporal gene expression data from yeast cells. (See Figure 1.1b.) In this case, for each gene, the expression level is observed at a total of 18 time points over the course of roughly 2 hours. Once again, the grid of observed points is fixed, but unlike the Canadian weather data, there are many fewer observations. In this case, the underlying process must be estimated through some process, such as smoothing splines.

In the case of longitudinal data, even less information may be available. Particularly when the subjects involved are people, it may not be possible to observe the subject at regular intervals, and some times, subjects may only be observed as few as one or two times. Data of this type is referred to as sparse, irregularly sampled data, since times at which observations are made are not fixed and the number of observations for some subjects may be very few. This is typical of longitudinal studies such as the Baltimore Longitudinal Study of Aging, where values such as blood pressure were measured on subjects when they visited the Gerontology Research Center. (Shock et al., 1984) Since subjects would drop out of the study or miss appointments, not all individual trajectories were frequently observed or observed over the full time frame of the study. Similarly, Bachrach et al. (1999) observed spinal bone mineral densities for males and females from ages 9 to 25 years old. Samples were taken at irregular intervals and for

many of the subjects only a few observations were made. (In some cases, as few as one or two observations were recorded.) From Figure 1.1c, we can see that these data look quite different from the previous examples.

1.2.1 Regression for functional data

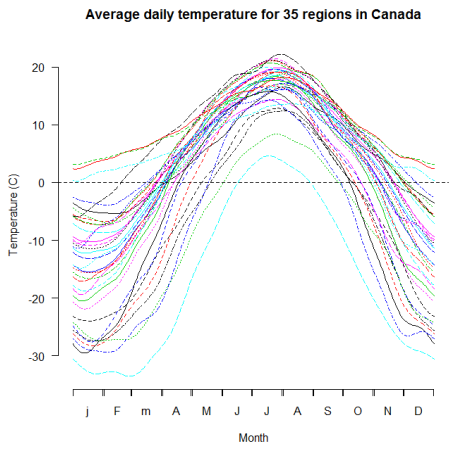
Consider the functional linear regression model

$$Y_i = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \epsilon_i, \qquad i = 1, \ldots, n. \tag{1.8}$$

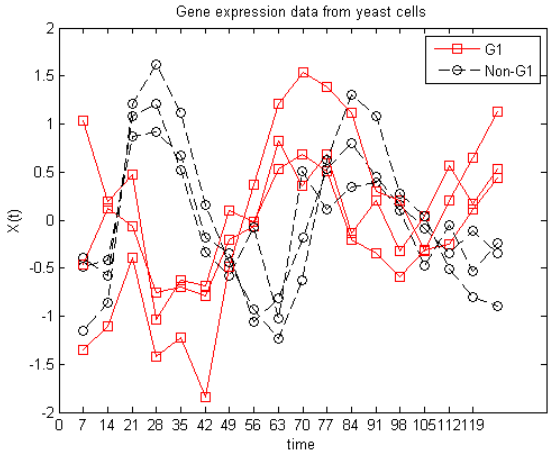
Here, the scalar response, Y_i is modeled using a functional predictor variable, $X_i(t)$, defined over some domain, $\mathcal{T} \in \mathbb{R}$. Often times for simplicity, $\mathcal{T}$ will be scaled to $[0, 1]$. The relationship between y_i and $X_i(t)$ is determined by the coefficient function, $\beta(t)$, and ϵ_i is random error. We assume that the predictor curves, $X_i(t)$ and the coefficient function, $\beta(t)$ are square-integrable functions on $\mathcal{T}$.

1.2.2 Notation for functional data

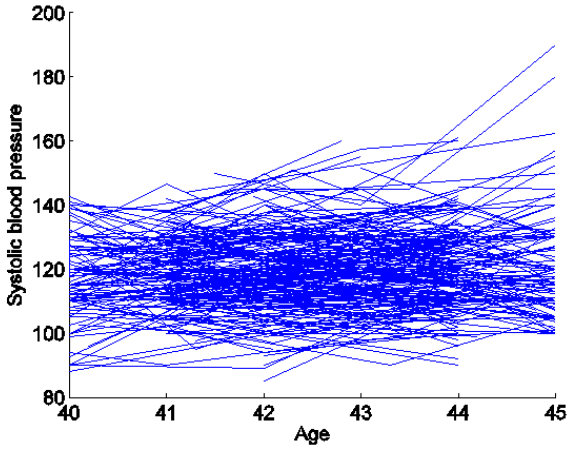
Typically, the full curve, $X_i(t)$ is not observed. Rather, for the ith curve, we observe $X(t)$ at some number $N_i > 0$ of discrete time points, $T_{i1}, ..., T_{iN_i}$ ordered such that $T_{ij} < T_{i(j+1)}$, where $T_{ij} \in \mathcal{T}$ and $j = 1, \ldots, N_i$. The number of observations per curve, N_i may be fixed so $N_i = N$ for all i or it may be randomly determined. In the case where the number of observations per curve is fixed, the time points of these observations may also be fixed (as in some genetics



(a) Canadian weather data



(b) Gene expression data



(c) BLSA data

Figure 1.1: Three types of functional data

studies, *e.g.* Rhein and Strimmer (2006) and Spellman et al. (1998)) or it may be randomly determined (*e.g.* James and Hastie (2001), Bachrach et al. (1999), Rice and Wu (2001)).

Additionally, we assume these observations are made with some measurement error, denoted δ_{ij}. Let the observed value for the ith curve at time T_{ij} be $U_{ij} = X_i(T_{ij}) + \delta_{ij}$. The measurement errors are assumed be independent both between and within curves with mean 0 and $var(\delta_{ij}) < \infty$. Our regression model requires the full curve, $X_i(t)$ to find Y_i. Thus, we must estimate the full curve using the available data before performing functional regression.

It is useful then to think of functional data as being one of two types. For the case where we have fixed time points, the straightforward approach of using a local smoother over the observed time points can be employed provided the grid of observations is sufficiently dense. (Ramsay and Silverman, 2005) Other approaches are necessary when the data is sparsely sampled at irregular time points. One such method is Principal component Analysis through Conditional Expectation or PACE, (Yao et al., 2005b) which can be applied to irregularly sampled data with very few observations per time point. The resulting estimated principal component functions and scores can be used to estimate the underlying curves, $X_i(t)$. We will discuss PACE further in the next section.

Once we have estimated the predictor curves, we perform regression and estimate the coefficient function. If we restrict $\beta(t)$ to the space of smooth functions with no further assumptions, identifiability is an insurmountable problem. One approach to dealing with this is to perform regression with a smoothness penalty. That is, we choose $\widehat{\beta}(t)$ to minimize

$$\sum \left(Y_i - \int_{\mathcal{T}} X_i(t)\widehat{\beta}(t)dt \right) + \lambda \int_{\mathcal{T}} \widehat{\beta}^{(d)}(t)dt, \tag{1.9}$$

where $\beta^{(d)}(t)$ is the dth derivative of $\beta(t)$ and λ is a penalty term. Thus, the estimated function is chosen to have smooth dth derivative. A more common method is to assume that the coefficient function can be decomposed using some orthogonal basis such that the first p basis functions can well-approximate $\beta(t)$. Then we write $\beta(t) = \boldsymbol{B}(t)^T\boldsymbol{\eta}$, where $\boldsymbol{B}(t) = [b_1(t), b_2(t), \ldots, b_p(t)]^T$, and $\boldsymbol{\eta}$ is a vector of coefficients. Thus, the regression problem becomes

$$Y_i = \beta_0 + \boldsymbol{x}_i^T\boldsymbol{\eta} + \epsilon_i, \tag{1.10}$$

where $x_{ij} = \int_{\mathcal{T}} X_i(t)b_j(t)dt$. We can now find estimates for $\boldsymbol{\eta}$ using any number of traditional methods, such as OLS described above. When we employ this basis decomposition approach, the choice of orthogonal basis, $\boldsymbol{B}(t)$ is of paramount importance. If we use PACE to estimate the curves, the principal component basis makes the most sense. In this case, the FPC scores become the design matrix, X.

1.2.3 Functional Principal Components

Functional Principal Components (FPC) is a useful method for analyzing functional data. Recall that for standard data, principal components is a nonparametric method used for exploratory data analysis and dimension reduction. For the regression model defined in Eq. 1.2, let X be a normalized design matrix. Then $(X^TX)^{-1}$ is the corresponding scaled covariance matrix. We can find the principal components by calculating the eigenvectors from X^TX. Note that these eigenvectors will be mutually orthogonal and form a complete basis for the vector space defined by X^TX. Equivalently, we can define the kth eigenvector to be the vector, $\boldsymbol{g}_k$

that solves

$$\underset{x}{\text{argmax}}\, \boldsymbol{g}\boldsymbol{g}_l^T(X^TX)\boldsymbol{g}_l, \;\; s.t. \;\; \boldsymbol{g}_l^T\boldsymbol{g}_l = 1 \text{ and } \boldsymbol{g}_l^T\boldsymbol{g}_m = 0 \; \forall m < l. \tag{1.11}$$

That is, the lth principal component will be the unit vector in the direction of maximal variation for X^TX that is orthogonal to the first $l-1$ eigenvectors.

Once we have found the eigenvectors, we can calculate the principal component scores for each observation by finding linear combination of the eigenvectors that returns its original coefficients. For each eigenvector, there is an associated eigenvalue, λ_l, which corresponds to the variability of X explained by the eigenvalue. It is standard to order the eigenvectors/eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. The dimension of the problem can be reduced by choosing a subset of eigenvectors, $l = 1, \ldots, K$ such that $\sum_{l=1}^{K} \lambda_l$ is sufficiently close to $\sum_{l=1}^{p} \lambda_l$. The reduced model is then

$$\boldsymbol{y} = X^D\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \tag{1.12}$$

where X^D is the matrix of the first K principal component scores for each observation, and $\boldsymbol{\beta}^*$ is the coefficient vector for the eigenvectors.

Functional principal components can be defined through analogy to the standard principal components method described above. Under the functional regression model given in Eq. 1.8, the predictor curves, $X_i(t)$ have an associated covariance operator, $V(s,t)$. Then the lth functional principal component (equivalently, lth eigenfunction) can be defined as

$$\underset{\phi_l(t)}{\operatorname{argmax}} \int_{\mathcal{T}}\int_{\mathcal{T}} \phi_l(s)V(s,t)\phi_l(t)dsdt \;\; s.t. \;\; \int_{\mathcal{T}} \phi_l(t)\phi_l(t)dt = 1, \;\; \int_{\mathcal{T}} \phi_l(t)\phi_m(t)dt = 0 \,\forall\, m < l. \tag{1.13}$$

Then, the lth eigenfunction is the normalized function that captures the most variability from $V(s,t)$ and is orthogonal to all previous eigenfunctions. Unlike in the traditional principal components case, this is not a finite basis, and $l = 1, 2, \ldots$. Typically, this expansion is truncated after K eigenfunctions such that $\sum_{l=1}^{K} \lambda_l$ is sufficiently close to $\sum_{l=1}^{\infty} \lambda_l$, where the lth eigenvalue, λ_l is defined as

$$\lambda_l = \int_{\mathcal{T}}\int_{\mathcal{T}} \phi_l(s)V(s,t)\phi_k(t)dsdt. \tag{1.14}$$

We can write our data using the Karhunen-Loève representation and the FPC basis as

$$X_i(t) = \mu(t) + \sum_{l=1}^{\infty} \xi_{il}\phi_l(t), \tag{1.15}$$

where $\mu(t)$ is the mean function of $\{X_i(t)\}_{i=1}^{n}$, ξ_{il} is the lth FPC score for the ith curve, and $Var(\boldsymbol{\xi}_l) = \lambda_l$.

1.2.4 FPC for Sparse and Irregularly Sampled Data

Many examples are given in the literature for finding the eigenfunctions, eigenvalues, and FPC scores. For example, see Castro et al. (1986) and Ramsay and Silverman (2005). However,

these methods are focused on the case when the predictor curves are sampled over a dense grid. When data is sparsely sampled at irregular intervals, finding consistent estimators for the FPC scores can be challenging. Principal components Analysis through Conditional Expectation (PACE, Yao et al. (2005b)) offers a way to find these estimates when some curves are observed as few as one or two times.

Suppose instead of observing the predictor curves on a dense grid they are observed sparsely and irregularly, as in the case of the BLSA data from Figure 1.1c. In this scenario, curve $X_i(t)$ is observed at time points $T_{i1}, ..., T_{iN_i}$ with measurement error δ_{ij}, such that at time T_{ij} we observe $U_{ij} = X_i(T_{ij}) + \delta_{ij}$. Let $\boldsymbol{T}_i$ and $\boldsymbol{U}_i$ denote the vectors of time points and observed values for the ith curve respectively. Using local polynomial smoothers (see, for example, Fan and Gijbels (1996)), we can estimate the mean function for the data, $\mu(t)$ as well as the covariance surface, $V(s,t)$. Consistent estimates for the lth FPC score for the ith curve conditional on the time points observed can be written as

$$\tilde{\xi}_{il} = E(\xi_{il}|\boldsymbol{U}_i, \boldsymbol{T}_i) = \lambda_l \boldsymbol{\phi}_{il}^T \Sigma_{U_i}^{-1}(\boldsymbol{U}_i - \boldsymbol{\mu}_i). \tag{1.16}$$

In this expression, $\Sigma_{U_i} = V(\boldsymbol{T}_i, \boldsymbol{T}_i)$, $\boldsymbol{\phi}_{il} = \phi_l(\boldsymbol{T}_i)$, and $\boldsymbol{\mu}_i = \mu(\boldsymbol{T}_i)$ respectively. These are all estimated using the estimated version of these processes found earlier. Along with the lth estimated eigenvalue, we can plug in these estimates to find our estimator for the (i,l)th FPC score. The PACE estimate for the ith curve is then

$$\widehat{X}_i(t) = \widehat{\mu}(t) + \sum_{l=1}^{K} \widehat{\xi}_{il}\widehat{\phi}_l(t), \tag{1.17}$$

where K is chosen via some algorithm. (BIC is suggested by Yao et al. (2005b).) Figure 1.2 shows the gene expression data discussed above as well as the estimated curves using PACE.

1.2.5 Fitting Functional Regression Models

Once the true curves are estimated, the regression model must be fit. The functional linear regression model (Yao et al. (2005a) and Ramsay and Dalzell (1991), for example) use the model

$$E(Y|X) = \beta_0 + \int_{\mathcal{T}} \beta(t) X^C(t) dt, \tag{1.18}$$

where $X^C(t) = X(t) - \mu_X(t)$ is the centered predictor function and $\beta(t)$ is the corresponding coefficient function. This coefficient function is analogous to the coefficient vector in a multiple linear regression, where instead of a functional predictor we have a vector of predictor variables associated with each response.

Much like linear regression in traditional settings, functional linear regression may be too restrictive to accurately model the relationship between the predictor and the response. Yao and Müller (2010) introduced functional quadratic regression and general functional polynomial regression which allow for more flexible fits. The functional quadratic regression model assumes

$$E(Y|X) = \beta_0 + \int_{\mathcal{T}} \beta_1(t) X^C(t) dt + \int_{\mathcal{T}} \int_{\mathcal{T}} X^C(s) \beta_2(s,t) X^C(t) ds dt, \tag{1.19}$$

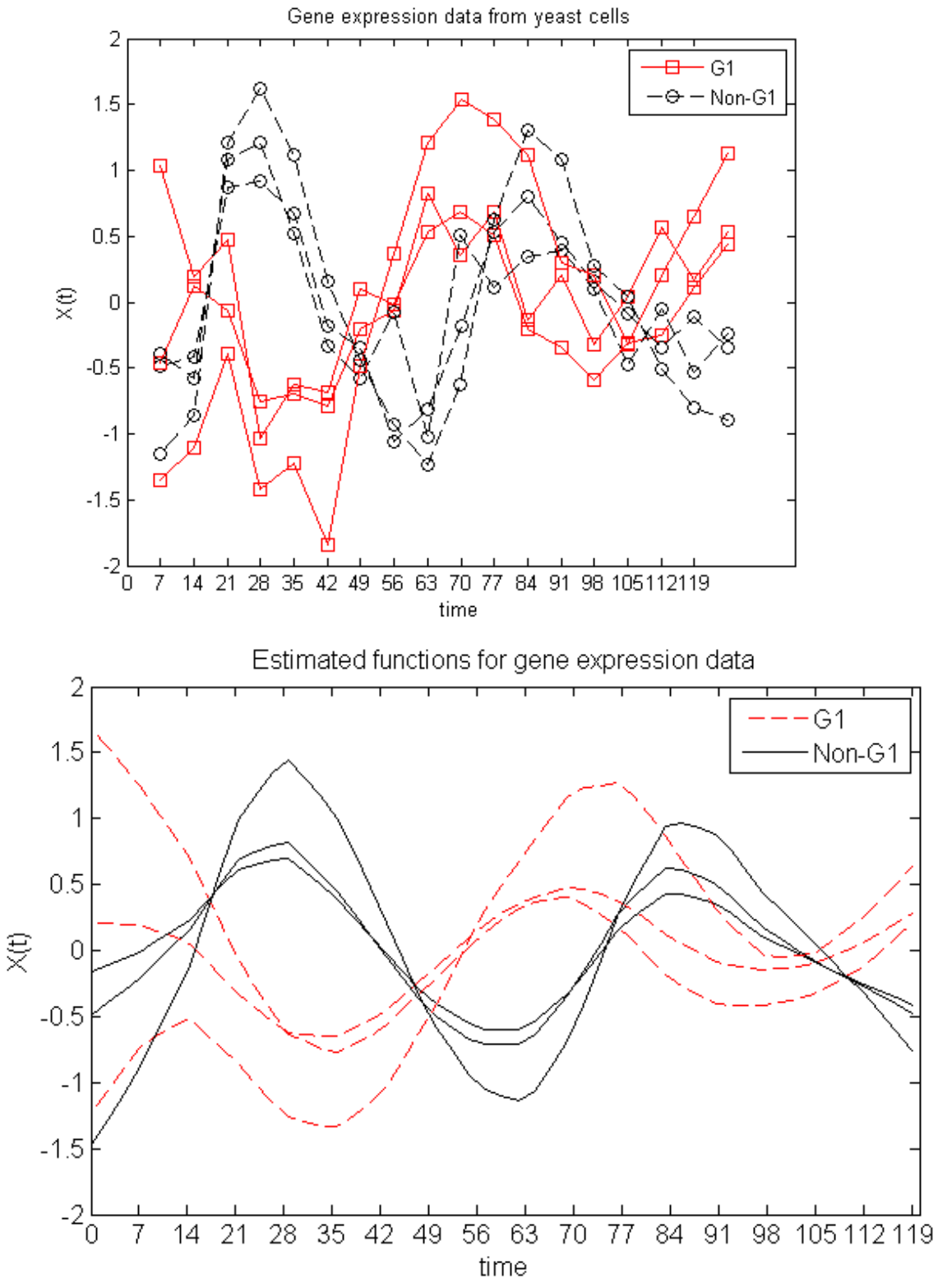


Figure 1.2: Gene expression data before and after smoothing with PACE

with similar extensions possible for higher-order terms. By including a quadratic term, this model is capable of accurately modeling a larger set of relationships between the predictor and response than linear regression.

The approach to estimation is similar for both of these models. First, an appropriate basis is chosen. Yao et al. (2005a) and Yao and Müller (2010) use PACE to estimate the functional principal component basis and corresponding FPC scores, but when full curves are observed, other bases are viable. For estimablity, we truncate the chosen basis after p basis functions. Denote this truncated basis as $\boldsymbol{B}(t) = [b_1(t), b_2(t), \ldots, b_p(t), ...]^T$. Then the coefficient function can be approximated as $\beta(t) = \boldsymbol{B}(t)^T\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is a vector of coefficients. Thus, the linear regression problem becomes

$$Y_i = \beta_0 + \boldsymbol{x}_i\boldsymbol{\eta} + \epsilon_i, \tag{1.20}$$

where $\boldsymbol{x}_{ij} = \int_{\mathcal{T}} X_i(t)b_j(t)dt$. We can estimate $\boldsymbol{\eta}$ using any of the traditional methods described above, such as OLS.

The process is similar for functional quadratic regression, with

$$\beta_2(x,t) = \sum_{m=1}^{p}\sum_{l=1}^{m}\eta_{lm}\phi_l(s)\phi_m(t)$$

. Using this, we can find a design matrix and solve using OLS or some other method.

1.3 Dissertation Outline

The remainder of this dissertation is organized in three main sections. Each section introduces a new method for analysis of functional data. In Chapter 2, we discuss sparse functional linear regression. The problem of interest is to identify regions on the domain, $\mathcal{T}$ of the functional predictor, $X(t)$, such that $\forall\ t \in \mathcal{T}' \subset \mathcal{T}$, $\beta(t) = 0$. We propose a two stage method that uses the Fused LASSO with a 1st order b-spline basis to identify these regions. This method is flexible in that it can be applied after any method is used to find an initial (not necessarily sparse) fit.

Chapter 3 addresses the problem of classification for functional data. There are examples in the literature (including the spinal bone mineral density data discussed above) where curves come from two or more classes. In such cases, identifying the differences between these classes and being able to classify new curves is useful. We propose a method motivated by linear discriminant analysis to address these issues. Our method is generally simpler in its application than other methods, can be generalized to cases with three or more classes, and is computationally efficient. Additionally, it can be applied to sparse and irregularly sampled data since it is based on PACE.

Finally, Chapter 4 introduces a method for nonlinear functional regression. Functional linear regression and functional quadratic regression have both been discussed in the literature, but both methods rely on parametric models. If the relationship between the predictor and response do not conform to the assumed model, these methods can produce poor results. Other nonparametric models for functional data exist in the literature, but these assume that the predictor curve is fully observed. Our method does not rely on parametric model assumptions, so it is more flexible than even functional polynomial regression, and it can be used with sparse and

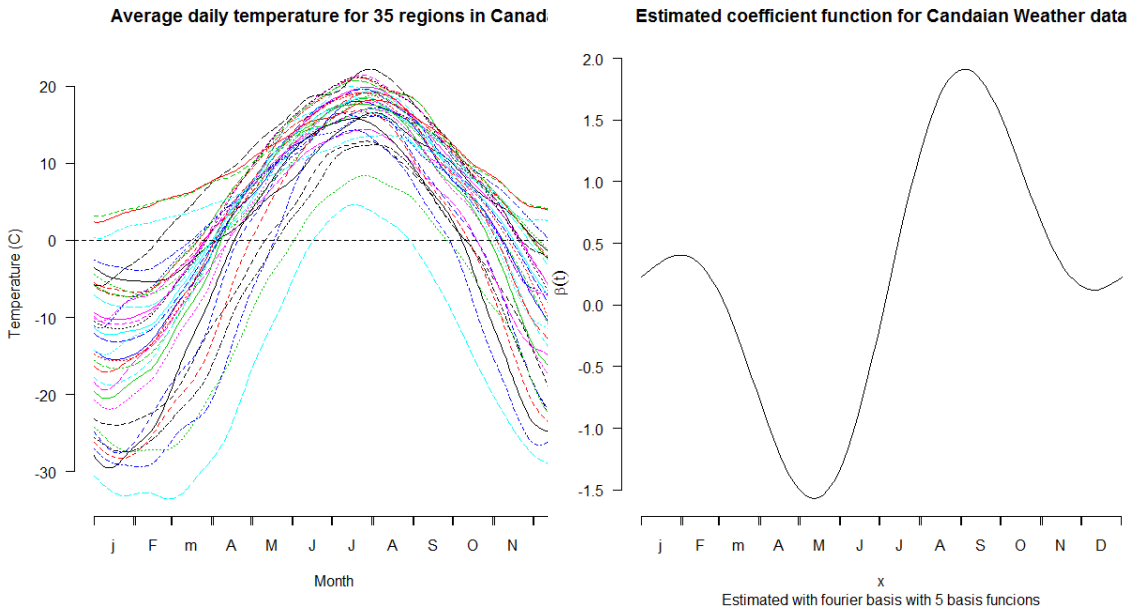
irregularly sampled data.

Chapter 2

Model Selection for Functional Data

2.1 Motivation

Model selection is an important area in traditional statistical analysis. Methods such as the LASSO are used to choose models that simultaneously capture the relationship between the response and covariates and ensure a sparse fit. These two characteristics are also desirable when considering functional data, but the notion of "sparsity" may look somewhat different in this setting. When multiple covariates are considered, analysis similar to the scalar case may be appropriate, but for a single functional predictor, "sparsity" can viewed relative to the domain of the covariate. A"sparse" model can be defined as one with a coefficient function, $\beta(t)$, that takes the value zero over a subset of the domain, $\mathcal{T}$, of $X(t)$. That is for some $\mathcal{T}' \subset \mathcal{T}$, $\beta(t) = 0 \; \forall t \in \mathcal{T}'$. Typically, we assume $\mathcal{T}'$ consists of a small number of contiguous subintervals of $\mathcal{T}$. This makes sense from a theoretical point of view (the alternative would be a highly erratic coefficient function) and will make the results easier to interpret.



(a) Average daily temperatures at 35 different weather stations

(b) Estimated coefficient function for Canadian weather data

Figure 2.1: Canadian Weather Data

For example, consider the Canadian weather data discussed in Ramsay and Silverman (2005). Average daily temperatures over nearly four decades are treated as predictors for annual rainfall in 35 distinct regions in Canada. A plot of the predictor curves is given in Figure 2.1a. Using standard functional regression techniques, we can estimate a coefficient function relating the daily temperatures to the annual rainfall of a region. Figure 2.1b shows an estimate for $\beta(t)$ using functional linear regression with a fourier basis. This fit shows two major peaks (one in the spring, the other in the fall) but does not identify any regions where there is no relationship between average daily temperature and rainfall. A sparse fit would ideally retain as non-zero the regions where the relationship between temperature and annual rainfall was strongest but also identify other regions where the relationship may not be relevant.

The remainder of this chapter is organized as follows. In Section 2, we discuss methods for

functional regression and existing methods for sparse functional regression. Section 3 introduces our new method for sparse functional regression, and Section 4 discusses computational issues. We demonstrate the effectiveness of this method through simulation in Section 5 and show the performance of our method on the Canadian weather data set. Section 6 concludes with a brief discussion.

2.2 Functional Regression

2.2.1 General Approach

A standard approach to functional regression is to assume that the coefficient function can be decomposed using some orthogonal basis such that the first p basis functions can well-approximate $\beta(t)$. Then we write $\beta(t) = \boldsymbol{B}(t)^T\boldsymbol{\eta}$, where $\boldsymbol{B}(t) = [b_1(t), b_2(t), \ldots, b_p(t)]^T$ and $\boldsymbol{\eta}$ is a vector of coefficients. Thus, the regression problem becomes

$$Y_i = \beta_0 + \boldsymbol{x}_i\boldsymbol{\eta} + \epsilon_i, \tag{2.1}$$

where $\boldsymbol{x}_{ij} = \int_{\mathcal{T}} X_i(t)b_j(t)dt$. We can now estimate $\boldsymbol{\eta}$ using any number of traditional methods, such as OLS described above. When we employ this basis decomposition approach, the choice of orthogonal basis, $\boldsymbol{B}(t)$ is of paramount importance.

This is especially true for the problem of sparse regression, since our goal is to find a sparse fit. An intuitive approach is to break $\beta(t)$ into a collection of basis functions defined on disjoint subsets of $\mathcal{T}$. We can then use traditional variable selection to choose which of these basis

functions should be included in the model (or equivalently, which should have their coefficients set equal to 0). B-splines are well-suited for this, since they are only defined on relatively small intervals on $\mathcal{T}$. For the simplest case, we can assume that $\beta(t)$ is a step function and choose a 1st order b-spline basis to model it. Thus, we rewrite the regression function as

$$Y_i = \sum_{j=1}^{p} \beta_j \int b_j(t) X(t) dt, \tag{2.2}$$

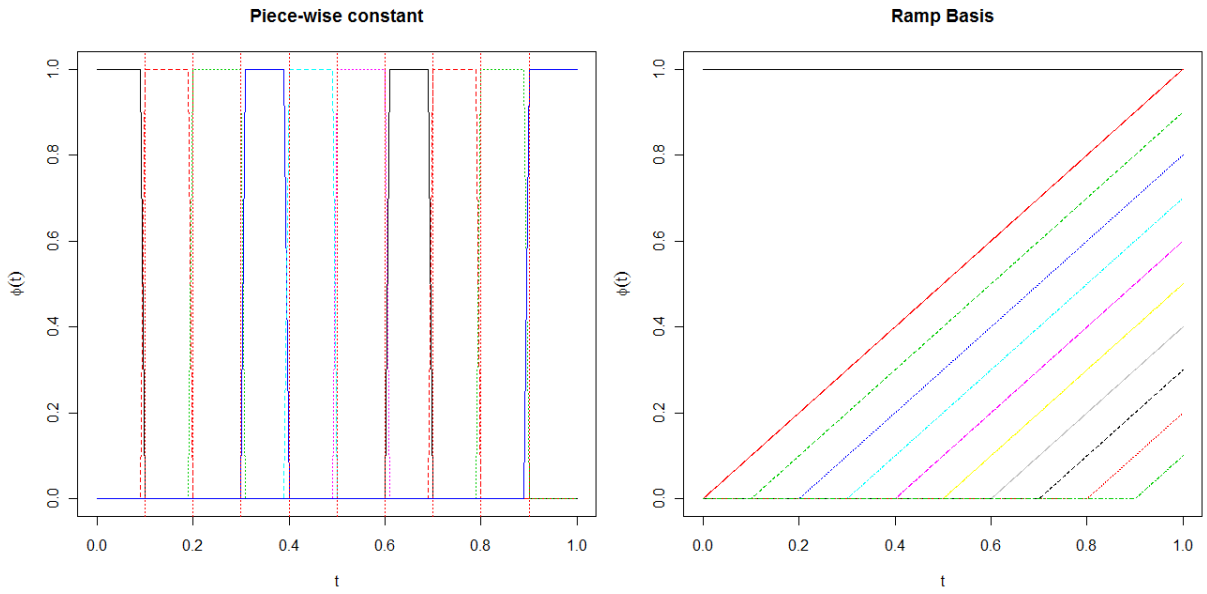
where

$$b_j(t) = I(t \in T_j). \tag{2.3}$$

Each $b_j(t)$ takes nonzero values only on the region T_j where $T_j = [t^*_{j-1}, t^*_j)$ for $j = 1, \ldots, p-1$ and $T_p = [t^*_{p-1}, t^*_p]$. While this basis may be used to find a sparse fit, step functions may not be flexible enough to accurately model the function. To fit more complex models, a linear fit may be desired. The "ramp" basis may be useful for such cases. We define the "ramp" basis as

$$b_j(t) = I(t \in [t^*_{j-1}, 1]) * (t - t^*_{j-1}). \tag{2.4}$$

Note that for the "ramp" basis, it is necessary to include an additional basis function, b_0 whose coefficient determines the value of $\beta(0)$. Also, the ramp basis does not have the property that $\beta_j = 0$ implies $\beta(t) = 0 | t \in T_j$. This means that sparsity in the β_j's will not necessarily produce sparsity in $\beta(t)$. These two sets of basis functions can be seen in Figure 2.2a and Figure 2.2b.



(a) 1st order b-spline basis

(b) Ramp basis

Figure 2.2: Two sets of basis functions

2.2.2 Existing Methods

Example of sparse functional regression are not prevalent in the literature. Functional Linear Regression That's Interpretable (FLiRTI, James et al. (2009)) was designed to find estimates of coefficient functions that were easy to interpret. The authors proposed to find estimated coefficient functions that did not make large, rapid shifts. As such, they restrict certain derivatives of the coefficient function to be sparse and estimate the restricted model using either the LASSO or Dantzig selector. (Both approaches are endorsed by the authors.) The derivative on which the restriction is placed can be thought of as a tuning parameter as well as a limit on the complexity of the estimated functional predictor.

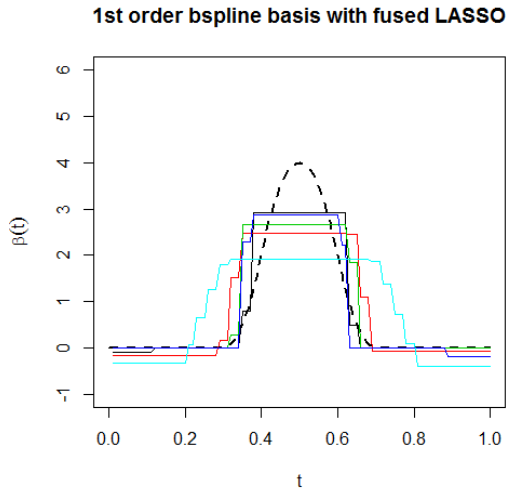
Lee and Park (2012) focus on limiting the number of basis functions included in the final model rather than finding large regions on $\mathcal{T}$ that are necessarily zero. As such, they arrive at sparsity indirectly. By choosing a b-spline basis with a high initial number of basis functions and then using any one of a variety of traditional variable selection techniques, they arrive at an estimated coefficient function that is zero for a substantial portion of its domain. While this method is novel in its assumption of a finite but unknown number of basis functions in the true model, it does not require the included basis functions to adhere to any sort of pattern. This could lead to fits that while sparse, may be difficult to interpret.

2.3 New Methods

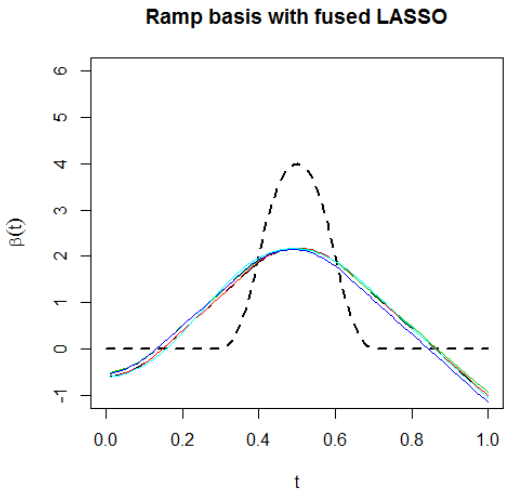
For vector data, one of the most common methods for model selection is the least absolute shrinkage and selection operator (LASSO, Tibshirani (1994)). The objective function for the lasso is

$$L_{\lambda_1}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\eta}) = .5(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\eta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\eta}) + \lambda_1 \sum_{j=1}^{p} |\eta_j|. \tag{2.5}$$

Since for our case, a natural ordering exists for our coefficients (they correspond to basis functions that represent consecutive subintervals on the domain of $X(t)$), the fused lasso (Tibshirani et al., 2005), which penalizes adjacent terms for "jumps", should also be considered. By "fusing" nearby terms, adjacent regions on the domain of $X(t)$ should be set to 0, making the results easy to interpret. The objective function for the Fused LASSO is



(a) 1st order b-spline fits using the Fused LASSO



(b) Ramp basis fits using the Fused LASSO

$$L_{\lambda_1,\lambda_2}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\eta}) = .5(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\eta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\eta}) + \lambda_1 \sum_{j=1}^{p} |\eta_j| + \lambda_2 \sum_{j=2}^{p} |\eta_j - \eta_{j-1}|. \tag{2.6}$$

The Fused LASSO takes advantage of the structure of the basis functions discussed above, all of which have a natural ordering. For the 1st order b-spline basis, if $\beta_i = 0$ then $\widehat{\beta}(t) = 0$ for the subinterval associated with $\phi_i(t)$. Thus, by fusing together adjacent coefficients, we are forcing $\widehat{\beta}(t)$ to take constant values across subintervals of its domain. Since the lasso penalty tends to push individual coefficients towards 0, the result is large subintervals over which $\widehat{\beta}(t) = 0$, giving us the sparse result we desire. Estimated coefficient functions fit using the fused LASSO and 1st order b-spline basis on five simulated data sets are shown in Figure 2.3a.

Fits generated using the 1st order b-spline basis are not ideal. By construction, the fitted function must be a step function and will not be continuous. Moreover, if the true coefficient function is linear or quadratic, the 1st order b-spline basis may struggle to find an accurate estimate.

A more flexible basis may be able to do a better job. In the case of the "ramp" basis, fusing the coefficients results in a fit whose first derivative is constant over a subinterval. Estimated fits from five simulated data sets are shown in Figure 2.3b. While this can make for an easy-to-interpret function, it does not necessarily result in sparsity.

Full details for these fitting of these models is given in the Appendix.

2.3.1 Two-stage fitting

To find a model that will simultaneously be sparse while retaining the flexibility to estimate complex functions, we take a two-stage approach. The first stage will give us an initial fit that does a good job of modeling the function irrespective of sparsity. The second stage will alter this initial fit to add sparsity.

The first stage is to estimate a coefficient function as accurately as possible using any method. For example, first stage fit could be estimated using functional linear regression with the "ramp" basis or a fourier basis. Define this estimate as $\widetilde{\beta}(t)$. For this first step, we do not consider the sparsity of the fit, instead aiming to capture the shape of the true coefficient function. Using $\widetilde{\beta}(t)$, we generate a new design matrix, $\boldsymbol{X}^*$:

$$\boldsymbol{X}^*_{i,j} = \int_{\mathcal{T}} X_i(t)\widetilde{\beta}(t)b_j(t)dt, \tag{2.7}$$

where $b_j(t)$ is the j basis function from the 1st order b-spline basis.

We then minimize

$$L_{\lambda_1,\lambda_2}(\boldsymbol{y}, \boldsymbol{X}^*, \theta) = .5(\boldsymbol{y} - \boldsymbol{X}^*\theta)^T(\boldsymbol{y} - \boldsymbol{X}^*\theta) + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{j=2}^{p} |\theta_j - \theta_{(j-1)}|. \quad (2.8)$$

This is the fused lasso model from Eq. 2.6. We choose the penalty terms, λ_1 and λ_2 via cross-validation.

The second stage design matrix, $\boldsymbol{X}^*$, incorporates the information from the first stage fit. If the coefficient function was estimated accurately, then $\boldsymbol{\theta} = \mathbf{1}^T$ should do a good job of minimizing Eq. 2.8. If there are regions on $\beta(t)$ that can be set to 0 without hurting prediction, the second stage should identify them, giving us a sparse fit. And if the first stage fit is not scaled correctly, the second stage will modify it to give us a more accurate final result. The final 2-stage estimate is computed by taking the point-wise product of the two estimated coefficient functions

$$\widehat{\beta}(t) = \widetilde{\beta}(t) * \widetilde{\theta}(t), \quad (2.9)$$

where $\widetilde{\theta}(t) = \sum_{j=1}^{p} \widetilde{\theta}_j \phi_j(t)$, and the $\{\phi_j(t)\}$ is the 1st order b-spline basis. $\widehat{\beta}(t)$ will be set to exactly 0 over intervals where the second-stage selector has specified a region to be 0. By using this two stage approach, we utilize the sparsity from the Fused LASSO and 1st order b-spline basis while retaining the good fit from generated in the first stage. The result is that we have the flexibility to model the coefficient function accurately while still ending up with a sparse fit.

Also, note that this two stage technique can be applied to any first-stage fit estimate in order to find a sparse fit. This means that we can use standard functional regression methods to estimate

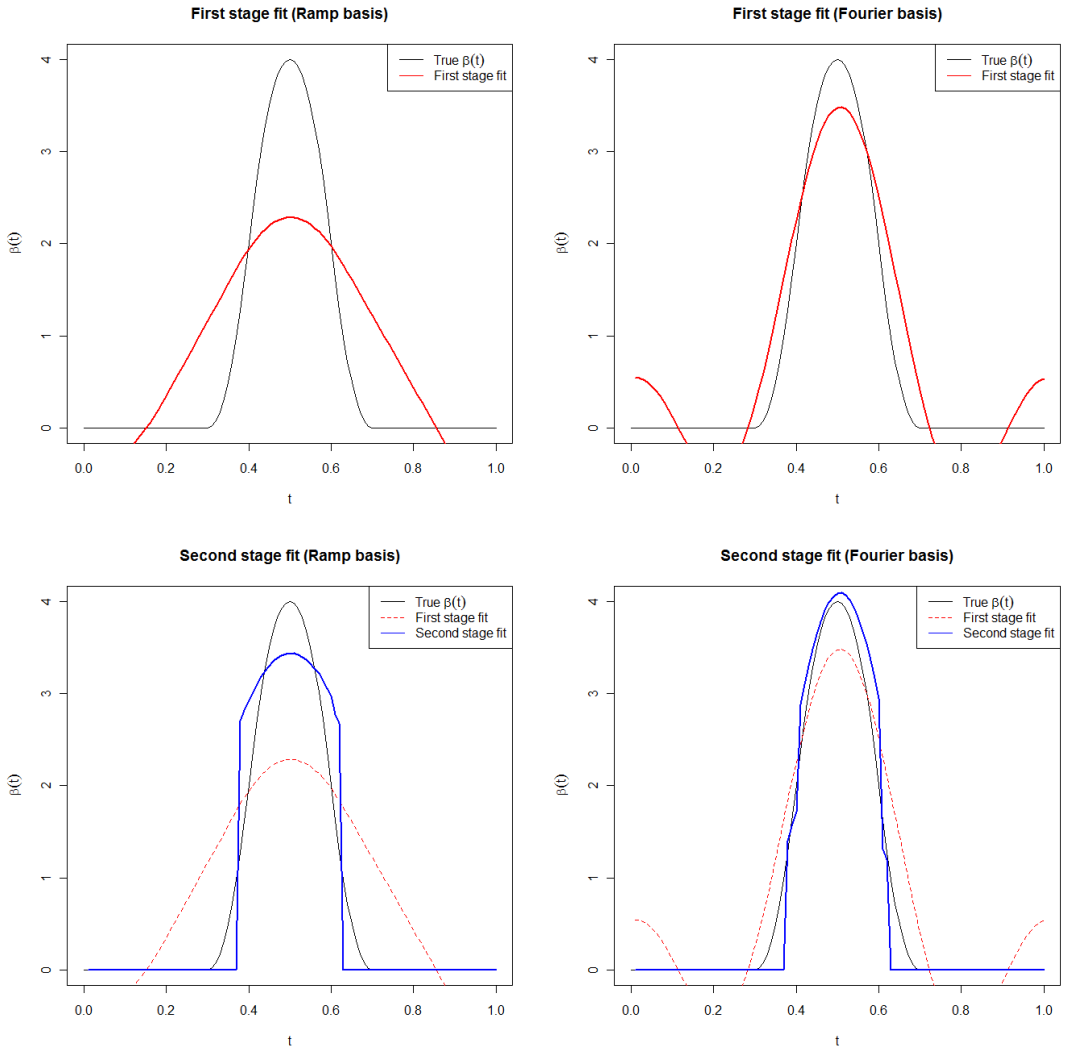
$\widetilde{\beta}(t)$ and then estimate $\widetilde{\theta}(t)$. Examples of this method using the "ramp" basis to estimate $\widetilde{\beta}(t)$ are given in Figure 2.3.

2.3.2 Sparsely Observed Trajectories and Measurement error

To this point, we haven't addressed how to estimate the predictor trajectories, $X_i(t)$. If the curves are fully observed, this is unnecessary. Consider the Canadian weather data. We observe each trajectory 365 times, making this a close approximation of the ideal situation when the full curve is observed. In many other applications, substantially less data may be available and some consideration must be given for finding estimates, $\widehat{X}(t)$ of the full curves.

When data is regularly observed on a relatively dense grid, it is possible to smooth each curve individually. In this case, many options are available. Various scatter plot smoothing methods (see Cleveland (1979) and Fan and Gijbels (1996) among others) as well as smoothing splines (Green and Silverman, 1994) are available for this type of analysis. When the data is sparse and irregularly sampled, estimating the individual curves can be more difficult. In such cases, it may be better to estimate all of the curves simultaneously. Principal components Analysis through Conditional Expectation (PACE, Yao et al. (2005b)) is one method for doing this. Once the principal component functions and scores have been estimated, estimates for the individual curves can be recovered.

Measurement error adds another level of complexity to the issue. Many of the issues associated with measurement error for vector data are also relevant to functional data. Particularly, while it may be possible to find unbiased estimates for each trajectory using smoothing splines as described above, when observations are made with measurement error, the coefficient function



(c) Ramp basis (d) Fourier basis

Figure 2.3: Initial and second stage stage fits

estimated with these curves, $\widehat{\beta}(t)$, will be biased. When the level of measurement error is very small, effect on the estimator may not be great, but with even modest levels of measurement error, the estimated coefficient function will be poor. In these cases, a calibration step should be used when estimating the predictor curves. Zhang et al. (2007) describe a nonparametric calibration approach which could be used to estimate the predictor trajectories. PACE (Yao et al., 2005b) is another possibility.

2.3.3 Summary of Algorithm

We summarize our method in three steps:

- Step 0: If necessary, find good estimates for the predictor curves, $\widehat{X}_i(t)$.
- Step 1: Obtain a good estimate for $\beta(t)$. (This can be done using any method; sparsity it not a concern.) Denote this as $\widetilde{\beta}(t)$
- Step 2: Create $\boldsymbol{X}^*$ and estimate sparsity function, $\widetilde{\theta}(t)$.
- Step 3: Combine $\widetilde{\theta}(t)$ and $\widetilde{\beta}(t)$ to obtain final, sparse estimate, $\widehat{\beta}(t)$.

2.4 Computational issues

2.4.1 Tuning for λ_1 and λ_2

The lasso and fused lasso penalties, λ_1 and λ_2 respectively, (denoted together as $\boldsymbol{\lambda}$) must be tuned appropriately to get a good fit. For standard penalized regression methods, many criterion for tuning penalty terms exist. In early work, we considered four methods including Akaike Information Criterion (AIC, Akaike (1974)), Bayesian Information Criterion (BIC, Schwarz (1978)), Extended Bayesian Information Criterion (EBIC, Chen and Chen (2008)), and cross-validation (CV). Each has its own unique objective function which we minimize to find "optimal" values for $\boldsymbol{\lambda}$. For cross-validation, we choose $\boldsymbol{\lambda}$ to minimize the cross-validation squared prediction error. The squared prediction error for the ith observation is

$$(Y_i - \widehat{Y}_i^{\boldsymbol{\lambda}})^2, \tag{2.10}$$

where $\widehat{Y}_i^{\boldsymbol{\lambda}}$ is the predicted response for the ith curve for a given value of $\boldsymbol{\lambda}$. Let $F_1, F_2, \ldots, F_K$ be randomly selected sets such that $F_1 \cup F_2 \cup \ldots \cup F_K = \{1, 2, \ldots, n\}$ and $F_k \cap F_m = \emptyset$, where $1 \leq m \neq k \leq n$. Let $F_k^C = \{1, 2, \ldots, n\}/F_k$ be the compliment of F_k. Then for a given $\boldsymbol{\lambda}$, the K-fold cross-validation squared prediction error is

$$Error_{\boldsymbol{\lambda}} = \sum_{k=1}^{K} \sum_{i \in F_k} (Y_i - \widehat{Y}_{F_k^C,i}^{\boldsymbol{\lambda}})^2, \tag{2.11}$$

where $\widehat{Y}_{F_k^C,i}^{\boldsymbol{\lambda}}$ denotes the predicted response for the ith curve based on the training data set, F_k^C,

and the smoothing parameter, $\boldsymbol{\lambda}$.

Alternatively, we can choose λ based on any of a few popular tuning criteria. For example, for AIC, we choose $\boldsymbol{\lambda}$ to maximize

$$AIC = nlog(\widehat{\sigma}^2) + 2df, \tag{2.12}$$

where $\widehat{\sigma}^2 = \sum(Y_i - \widehat{Y}_i)^2$ and df is the model degrees of freedom, defined as the number of unique values taken by the $\widehat{\beta}$ in the fused LASSO regression. Similarly, for BIC and EBIC, we have

$$BIC = nlog(\widehat{\sigma}^2) + log(n)df, \tag{2.13}$$

and

$$EBIC = nlog(\widehat{\sigma}^2) + (log(n) + log(p))df. \tag{2.14}$$

These two methods increase the penalty on using more degrees of freedom when the sample size becomes large with the objective of preventing over-fitting. In the case of EBIC, models that include larger numbers of basis functions are also penalized for overfitting. This criterion was originally devised with genome-wide association studies in mind. In that situation, overfitting can lead to high false discovery rates. This can be thought of as analogous to our situation, where we are hoping to limit the number of basis functions (each representing a region on $\mathcal{T}$) identified as non-zero so that our fit is sparse.

To choose $\boldsymbol{\lambda}$, the relevant objective function is optimized using a Newton-related method in the function *optim* with R (R Development Core Team, 2011). Based on early simulation results not reported here, cross-validation, BIC, and EBIC are roughly equivalent, with each showing marginally better performance in some situations. Optimization through cross-validation takes substantially longer than the alternatives. For the results reported below, EBIC was used to tune $\boldsymbol{\lambda}$.

2.4.2 Numerical integration and other issues

Regardless of whether the curves are fully observed or estimated by smoothing or another method, it is necessary to integrate the basis functions so that the design matrices can be populated. For numerical integration, we use Gaussian quadrature through the *gauss.quad.prob* function in the *statmod* package (with contributions from Yifang Hu et al., 2011). For each integration, 200 evaluation points are used. The primary reason for using quadrature over the *integrate* function is computation speed. When we treat the full curves as observed (in other words, we assume no measurement error and perform no smoothing), populating the design matrix is straightforward for simulated data. If we generate curves, $X_i(t) = \sum_{k=1}^{K} \xi_{ik}\phi_k(t)$, for the basis $\{b_j(t)\}_{j=1}^{p}$, the ijth element of the design matrix is

$$\boldsymbol{X}_{ij} = \int_{\mathcal{T}} X_i(t)b_j(t)dt \qquad = \int_{\mathcal{T}} \sum_{k=1}^{K} \xi_{ik}\phi_k(t)b_j(t)dt = \sum_{k=1}^{K} \xi_{ik} \int_{\mathcal{T}} \phi_k(t)b_j(t)dt. \qquad (2.15)$$

Since both $b_j(t)$ and $\phi_k(t)$ are known, only $K \times p$ integrations are necessary. For the second stage, Eq. 2.7 can be rewritten as

$$\boldsymbol{X}^*_{i,j} = \int_{\mathcal{T}} X_i(t)\widetilde{\beta}(t)b_j(t)dt, = \sum_{k=1}^{K} \xi_{ik} \int_{\mathcal{T}} \phi_k(t)\widetilde{\beta}(t)b_j(t)dt, \tag{2.16}$$

which once again only requires solving $K \times p$ integration problems.

When the predictor curves are estimated, each one must be treated individually and, we must solve

$$\boldsymbol{X}_{ij} = \int_{\mathcal{T}} \widehat{X}_i(t)b_j(t)dt \tag{2.17}$$

for each observation. This requires $i \times K \times p$ steps, requiring substantially more computation time. The same issue occurs when finding the matrix for the second stage estimate. (Note that this is less of a problem when the predictor curves are estimated using PACE, since each curve is written using an estimated principal component basis of finite dimension. Quadrature speeds up this process substantially, and the high number of evaluations ensures that the integral is accurate.

Finally, the Fused LASSO is fit using quadratic programming. The specifics of how the the Fused LASSO can be written as a quadratic programming problem are given in the Appendix. In R, solutions to quadratic programming problems can be found using the *solve.QP* function in the *quadprog* package (original by Berwin A. Turlach R port by Andreas Weingessel, 2011). For the simulations with measurement error, the curves are estimated using *smooth.spline* in *R*. (R Development Core Team, 2011)

2.5 Simulations

We discuss two cases here. In the first case, the data was treated as if the full curve was observed and there was no measurement error. This is analogous to the Canadian weather data example discussed earlier. Since the entire curve was observed, the data was not smoothed. To generate the data, we used five basis functions:

$$\phi_0(t) = 1 \tag{2.18}$$

$$\phi_1(t) = \sin(\pi t) \tag{2.19}$$

$$\phi_2(t) = \cos(\pi t) \tag{2.20}$$

$$\phi_3(t) = \sin(2\pi t) \tag{2.21}$$

$$\phi_4(t) = \cos(2\pi t), \tag{2.22}$$

where $X_i(t) = \sum_{k=0}^{4} \xi_{ik} f_k(t)$. The ξ_{ik} are independent and normally distributed with $\boldsymbol{\mu} = [0, 0, 0, 0, 0]^T$ and variance 4^2 for each coefficient. The recorded response variables are

$$Y_i = \sum_{k=0}^{4} \xi_{ik} \int_{\mathcal{T}} \phi_k(t)\beta(t)dt + \epsilon_i, \tag{2.23}$$

where $\mathcal{T} = [0, 1]$ and $\epsilon_i \approx N(0, \sigma^2 = 1^2)$. The coefficient function, $\beta(t)$ is one of the six coefficient functions shown in Figure 2.4.

For the second case, we looked at data where the full curve was not completely observed and

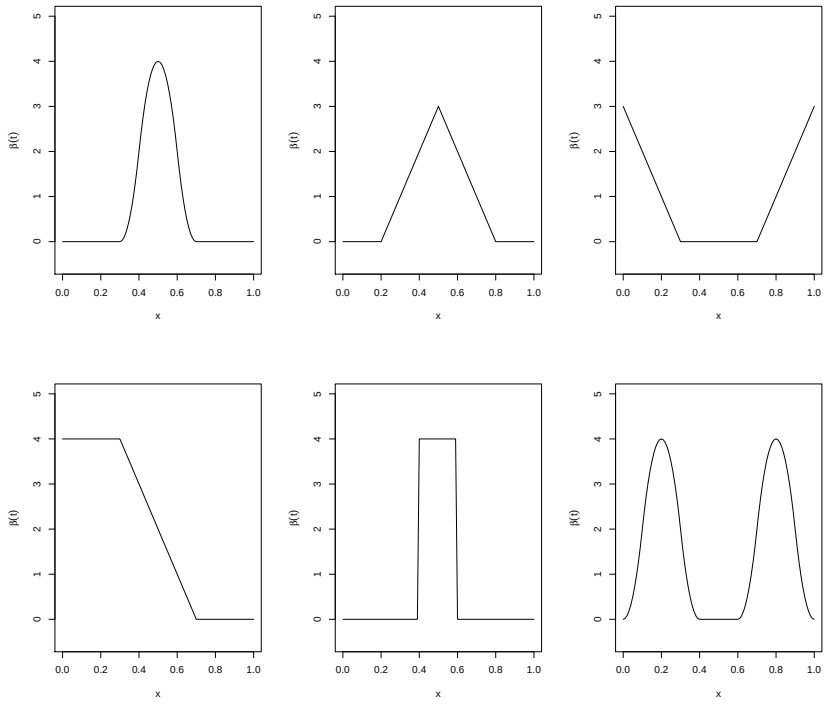
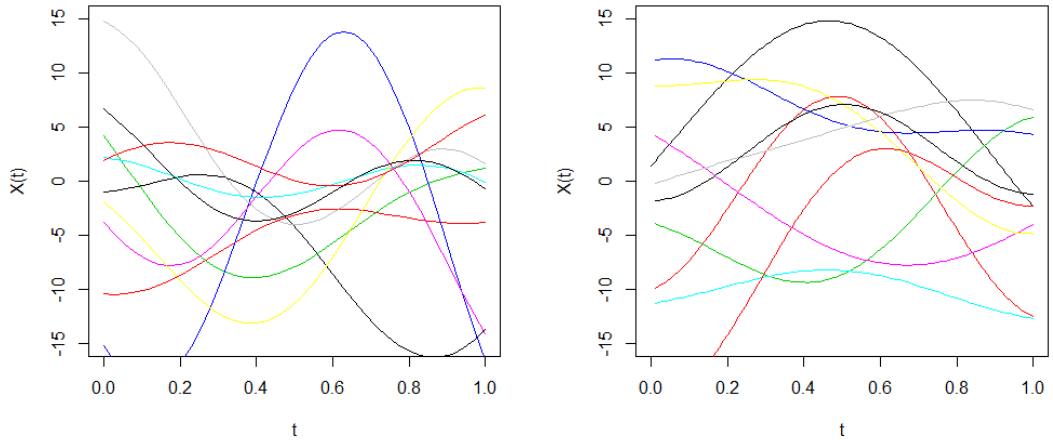


Figure 2.4: Unimodal, Triangle, Valley, Z, Step, and Bimodal coefficient functions (from left to right starting with the top row)

measurement error was present. For this case, we observed 36 time points on a fixed grid such that $T_j = \frac{j-1}{35}, \quad j = 1, \ldots, 36$. At each point, we observe $U_{ij} = X_i(T_j) + \delta_{ij}$, where δ_{ij} is measurement error with mean 0 and variance $var(\delta_{ij}) = \gamma^2$. We then smooth each curve using the function *smooth.spline* in *R*. (R Development Core Team, 2011) As in the case where the full curve is observed, the response is calculated using the true curve.

2.5.1 Assessing the quality of a fit

In motivating this work, we described two goals. The first was to find a good estimate for the coefficient function. The most straightforward way to assess this is to measure how far from



(a) Without measurement error (b) With measurement error ($var(\delta) = 0.01^2$)

Figure 2.5: Example curves

the true coefficient function our estimate is. That is, measure the squared difference between $\beta(t)$ and $\widehat{\beta}(t)$. We denote this $MISE_1$ and define it as

$$MISE_1 = \int_{\mathcal{T}} \left(\beta(t) - \widehat{\beta}(t)\right)^2 dt. \tag{2.24}$$

While this tells us how close our estimate was to the true coefficient function, the utility of regression also lies in how accurate predictions from the fitted model are. Thus, we also require a good fit to produce accurate estimates of the response variable. Therefore, we define $MISE_2$ to be

$$MISE_2 = n^{-1} \sum_{i=1}^{n} \left(\int_{\mathcal{T}} \beta(t) X_i(t) dt - \int_{\mathcal{T}} \widehat{\beta}(t) X_i(t) dt \right)^2. \tag{2.25}$$

This is similar to the mean squared prediction error except instead of Y_i we use $\int_{\mathcal{T}} \beta(t)X_i(t)dt = Y_i - \epsilon_i$. So $MISE_2$ is equivalent to the squared prediction error after removing random error.

The second motivating criterion was to find a fit that was sparse and accurately identified regions $\mathcal{T}'$ on $\mathcal{T}$ where $\beta(t) = 0 \forall t \in \mathcal{T}'$. Ideally, $\forall t \in \mathcal{T}\ s.t. \beta(t) = 0$, we would want $\widehat{\beta}(t) = 0$ as well, and similarly, $\forall t \in \mathcal{T}\ s.t. \beta(t) \neq 0$, we would want $\widehat{\beta}(t) \neq 0$. We define Type I error rate to be the proportion of points on a fine grid where $\beta(t) = 0$ but $\widehat{\beta}(t) \neq 0$, and Type II error rate to be the proportion of points on a fine grid where $\beta(t) \neq 0$ but $\widehat{\beta}(t) = 0$. In other words,

$$TypeIError = p_1^{-1} \sum_{i \in \Omega_1} (1 - \mathbf{1}_{\widehat{\beta}(t_i)=0}) \tag{2.26}$$

$$TypeIIError = p_2^{-1} \sum_{i \in \Omega_2} (1 - \mathbf{1}_{\widehat{\beta}(t_i)\neq 0}), \tag{2.27}$$

where Ω_1 and Ω_2 are the set of indices where $\beta(t_i) = 0$ and $\beta(t_i) \neq 0$ respectively, and n_1 and n_2 are the total time points observed in each of these groups. Note that the Fused LASSO does not set points to exactly zero but rather to be numerically 0. Thus, we treat any point where $|\widehat{\beta}(t)| < 10^{-10}$ as a "zero". A total of 1000 grid points were used to calculate these error rates.

2.5.2 Simulation Results

For the case where each curve is fully observed, we consider estimates from a variety of methods. First, we look at the Fused LASSO fit using the 1st order b-spline basis. This can be viewed as a "baseline" sparse fit using the Fused LASSO. Next, we considered two two-stage estimators. Initial fits using the "ramp" basis with Fused LASSO penalty and a simple linear

regression with a Fourier basis were considered. For both of these, a second stage estimate was computed which added sparsity. Finally, we look at the FLiRTI fits with $d = 1, 2$ for comparison. The result is a total of seven models for each data set, five of which are sparse fits. (The first stage "ramp" and Fourier fits are non-zero everywhere except where they cross the y-axis.)

Six coefficient functions defined on $\mathcal{T} = [0, 1]$ are considered. These can be seen in Figure 2.4. For each coefficient function, we consider sample sizes of $n = 50, 100, 200, 500$. Example fits using $n = 50$ from the 1st order b-spline basis, second stage "ramp", second stage Fourier, and FLiRTI ($d = 2$) are given in Figure 2.6 and Figure 2.7. Table 2.1 shows results for all seven fits (including first stage fits that are not sparse) for the unimodal coefficient function with $n = 50$. As we can see from this table, the second stage improves over the first stage for $MISE_1$ and $MISE_2$ for both cases. The Type I error rate also improves, while the Type II error rate gets worse. This same trend was observed across all coefficient functions and all levels of n. As such, we do not give results for these first stage estimates in the summaries of results that follow. Additionally, the FLiRTI results with $d = 2$ were generally a bit better than for $d = 1$, so once again in the interest of space, we only report the FLiRTI results for $d = 1$.

Table 2.1: Simulation results for all methods with $\beta(t)$ Unimodal, $n = 50$, $p = 35$

Coefficient Function	MISE 1	MISE 2	Type I error	Type II error
Constant	0.51	0.09	0.12	0.12
Ramp 1	0.99	2.03	0.63	0.00
Ramp 2	0.33	0.08	0.16	0.18
Fourier 1	0.80	0.11	0.43	0.02
Fourier 2	0.61	0.09	0.21	0.19
Flrt 1	0.85	0.11	0.08	0.35
Flrt 2	0.59	0.11	0.18	0.26
Median SE	0.048	0.008	0.019	0.016

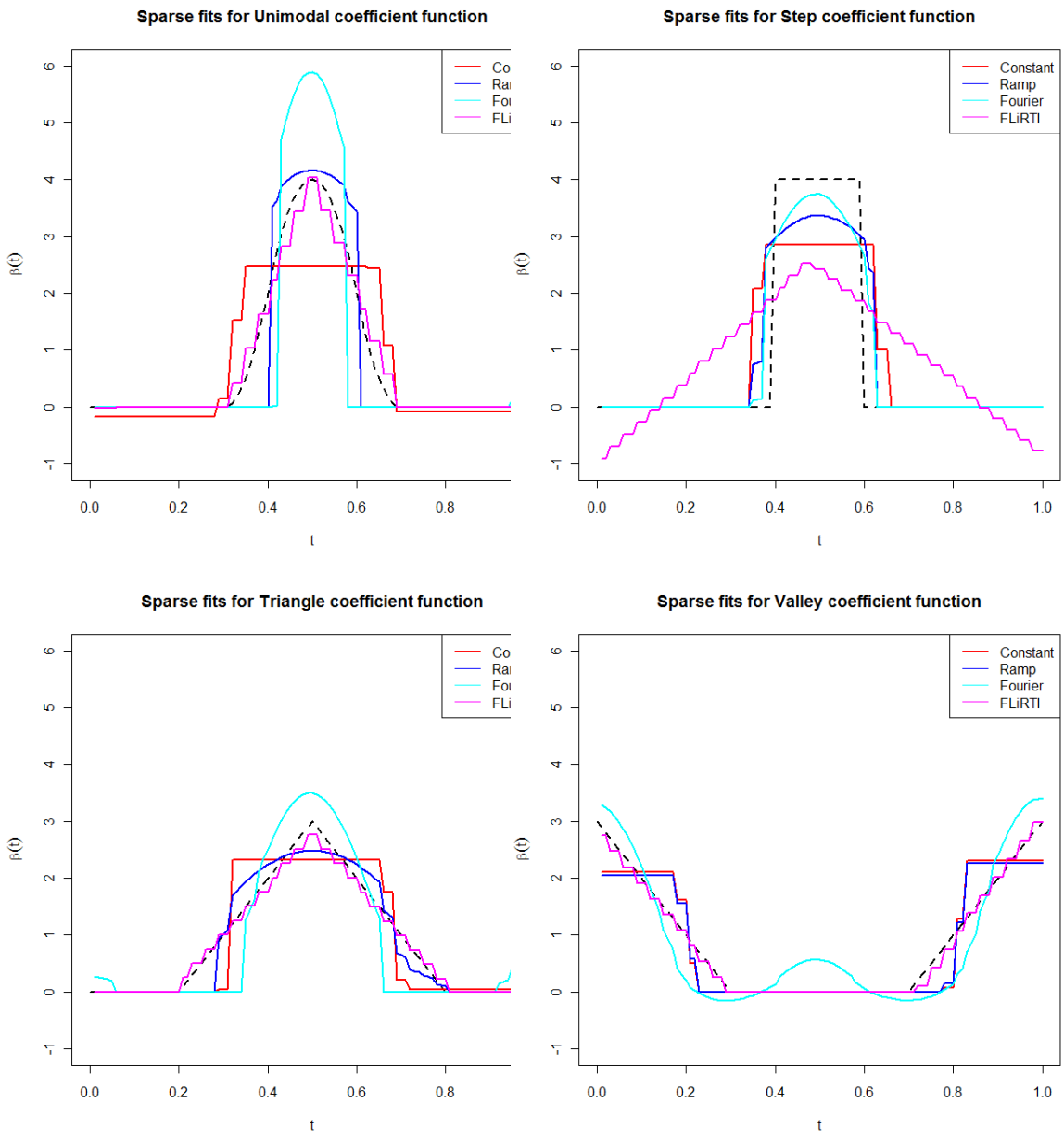


Figure 2.6: Sparse fits for different coefficient functions

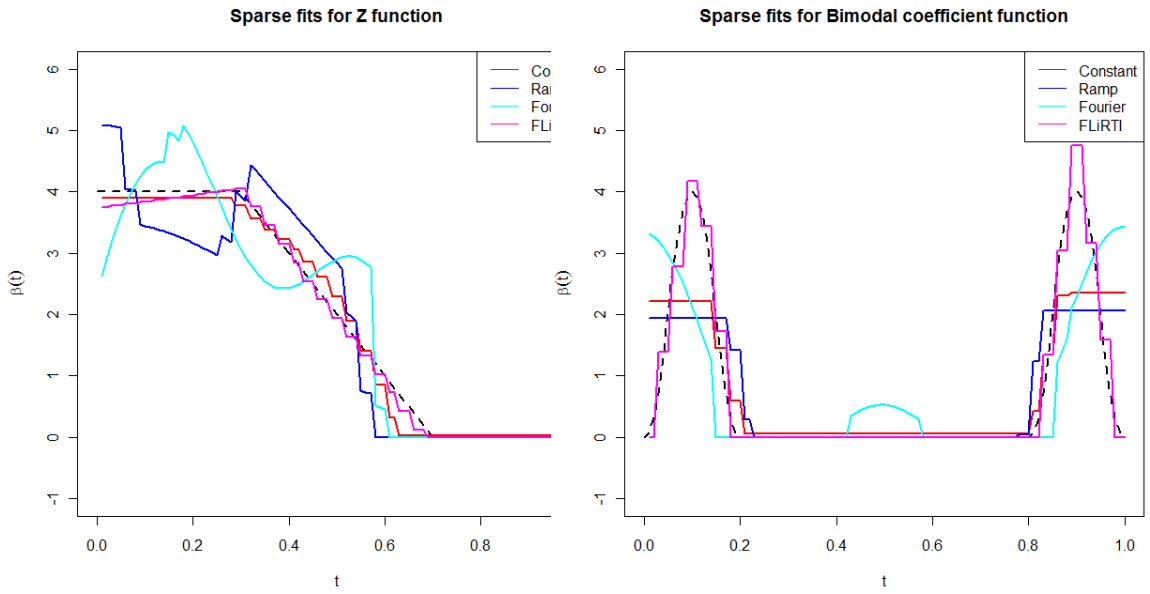


Figure 2.7: Sparse fits for different coefficient functions

From Table 2.2, we can see that the second stage ramp fit appears to be best overall when the sample size is smallest. As n increases from 50, however, the FLiRTI method improves quickly at identifying the zero-regions, while the two second stage fits from both the ramp and Fourier basis struggle. Indeed, we see the Type II error rates for both increase while the Type I error rates stay largely the same. The best overall method may be the single stage fit using the 1st order b-spline basis with the Fused LASSO. This method consistently has the lowest Type I error rate, and its Type II error rate is always first or second best. This comes at a price in terms of MISE 1, but MISE 2 for this method is competitive with the alternatives.

When we look at the fit for the Z coefficient function, (see Table 2.3) we again see the Fused LASSO methods generally outperforming FLiRTI in terms of MISE 1. This means that these methods are doing a better job of finding the shape of the Z coefficient function than FLiRTI. All methods are close in MISE 2, and no method stands out for selection error. The single stage

Table 2.2: Simulation results unimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	500
MISE 1	Piecewise Constant	0.51	0.43	0.41	0.32
	Ramp	0.33	0.31	0.24	0.22
	Fourier	0.61	0.38	0.31	0.17
	FLiRTI ($d = 2$)	0.59	0.41	0.27	0.20
	Median SE	0.044	0.037	0.035	0.023
MISE 2	Piecewise Constant	0.09	0.05	0.02	0.01
	Ramp	0.08	0.04	0.02	0.009
	Fourier	0.09	0.05	0.02	0.01
	FLiRTI ($d = 2$)	0.11	0.05	0.03	0.01
	Median SE	0.007	0.004	0.002	0.001
Type I error	Piecewise Constant	0.12	0.07	0.07	0.06
	Ramp	0.16	0.19	0.18	0.20
	Fourier	0.21	0.21	0.16	0.14
	FLiRTI ($d = 2$)	0.18	0.11	0.11	0.13
	Median SE	0.020	0.017	0.018	0.016
Type II error	Piecewise Constant	0.12	0.16	0.15	0.16
	Ramp	0.18	0.20	0.23	0.21
	Fourier	0.19	0.17	0.22	0.20
	FLiRTI ($d = 2$)	0.26	0.22	0.18	0.18
	Median SE	0.018	0.017	0.015	0.015

fit with the 1st order b-splines has substantially lower Type II error rates than the others, but it also has the worst Type I error rates. FLiRTI and the second stage ramp fit appear to find the best balance of the two types of error for the Z coefficient function.

Also, note that for this table, the results reported are the trimmed means for 3% trim on either side. This is due to the fact that the "ramp" basis produced some large outliers with $MISE_2$ error rates roughly 4 orders of magnitude larger than the other observations. Due to this exceptionally poor first order fit, the second order fit was similarly poor. This is indicative of a larger problem with the Fused LASSO combined with the "ramp" basis. Fits for this model appear

to be somewhat unstable and will occasionally produce large outliers such as those observed here. While we didn't find outliers like this with other coefficient functions, the standard errors for the ramp basis fits tended to be higher than other methods. The take away is that while the two stage fitting process can improve poor first order fits, if the initial fit is bad enough, the second stage fit will also be poor. More succinctly, "garbage in, garbage out."

Table 2.3: Simulation results for Z coefficient function for different values of n with EBIC tuning. ($p = 35$) Note: Trimmed means reported.

Error Type	Fit	n=50	100	200	500
MISE 1	Piecewise Constant	0.11	0.10	0.09	0.10
	Ramp	0.39	0.11	0.25	0.20
	Fourier	1.09	0.66	0.57	0.57
	FLiRTI ($d = 2$)	0.69	0.69	0.88	1.13
	Median SE	0.112	0.074	0.069	0.078
MISE 2	Piecewise Constant	0.11	0.05	0.02	0.01
	Ramp	1.48	0.05	0.02	0.01
	Fourier	0.12	0.05	0.02	0.01
	FLiRTI ($d = 2$)	0.11	0.06	0.03	0.01
	Median SE	0.012	0.005	0.002	0.001
Type I error	Piecewise Constant	0.31	0.34	0.48	0.55
	Ramp	0.24	0.15	0.24	0.19
	Fourier	0.12	0.06	0.06	0.07
	FLiRTI ($d = 2$)	0.19	0.15	0.13	0.13
	Median SE	0.033	0.033	0.033	0.033
Type II error	Piecewise Constant	0.06	0.05	0.03	0.03
	Ramp	0.10	0.08	0.06	0.07
	Fourier	0.19	0.10	0.22	0.21
	FLiRTI ($d = 2$)	0.08	0.07	0.10	0.12
	Median SE	0.06	0.08	0.06	0.08

The Bimodal coefficient function proves difficult for the Fused LASSO-based methods to model. This is to be expected, since these methods penalize large jumps in coefficient value, making bimodal functions problematic to estimate. From MISE 1, FLiRTI is clearly the best

method. However, MISE 2 is nearly equal across the board, and Type I and Type II error rates show divergent results, with FLiRTI having very low Type I error but correspondingly high Type II error, and the Fused LASSO based methods showing the opposite.

Results for the Step, Triangle, and Valley coefficient functions are given in the Appendix.

Table 2.4: Simulation results for bimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	1000
MISE 1	Piecewise Constant	0.88	0.85	0.81	0.84
	Ramp	0.94	0.86	0.84	0.78
	Fourier	1.42	1.04	0.92	0.89
	FLiRTI ($d = 2$)	0.86	0.75	0.63	0.61
	Median SE	0.038	0.033	0.023	0.021
MISE 2	Piecewise Constant	0.11	0.05	0.02	0.01
	Ramp	0.14	0.08	0.03	0.01
	Fourier	0.12	0.06	0.02	0.01
	FLiRTI ($d = 2$)	0.12	0.06	0.03	0.01
	Median SE	0.009	0.005	0.002	0.001
Type I error	Piecewise Constant	0.14	0.16	0.13	0.18
	Ramp	0.18	0.21	0.16	0.15
	Fourier	0.28	0.23	0.27	0.23
	FLiRTI ($d = 2$)	0.22	0.19	0.13	0.09
	Median SE	0.018	0.024	0.022	0.021
Type II error	Piecewise Constant	0.01	0.01	0.00	0.00
	Ramp	0.01	0.01	0.01	0.02
	Fourier	0.06	0.03	0.01	0.00
	FLiRTI ($d = 2$)	0.06	0.09	0.14	0.21
	Median SE	0.007	0.005	0.004	0.005

2.5.3 Sparely Observed Curves with Measurement Error

The previous results assumed the curves were fully observed, which is often unrealistic in functional data analysis. Here, we perform simulations where each curve is observed sparsely with small measurement error. On a grid of 36 fixed time points, we observe $U_{ij} = X_i(T_{ij}) + \delta_{ij}$, where $\delta_{ij} \approx N(0, \gamma^2)$. Below, we report results from simulations using the Unimodal and Z coefficient function with $\gamma = 0.01, 0.025$. We estimate each curve using a smoothing spline over the observed time points.

Simulations using the Unimodal coefficient function show that the FLiRTI ($d = 2$) method is probably the best of the four estimates looked at. (See Table 2.5.) For this coefficient function, only the single stage Fused LASSO fit with the 1st order b-spline basis was competitive. However, note that as the measurement error was increased from $\gamma = 0.01$ to 0.025, the difference between these two became closer, with the Fused LASSO performing better in $MISE_2$ and Type II error and FLiRTI performing better in the other categories.

Table 2.5: Error rates for Unimodal coefficient with measurement error

γ	Fit	MISE1	MISE2	Type I errror	Type II error
0.01	Fused Lasso	0.272	0.028	0.291	0.128
	2nd Stage Ramp	0.353	0.035	0.149	0.258
	FLiRTI 1	0.458	0.037	0.085	0.331
	FLiRTI2	0.268	0.041	0.166	0.158
0.025	Fused Lasso	0.361	0.037	0.362	0.123
	2nd Stage Ramp	0.540	0.059	0.207	0.273
	FLiRTI 1	0.622	0.065	0.161	0.303
	FLiRTI2	0.357	0.068	0.198	0.204

The story is different for the Z coefficient function. The results are displayed in Table 2.6. Here,

the Fused LASSO-based methods perform much better than both FLiRTI methods across the board. Between the two-stage "ramp" basis fit and the single stage 1st order b-spline fit, it is unclear which is the best. Both have similar $MISE_2$s, and while the single stage method is substantially better for $MISE_1$, its Type I error rate is much worse than the two stage alternative. Together with the results from the Unimodal coefficient function simulation, we can say that the two-stage Fused LASSO method works with some degree of success for data with measurement error. While its performance relative to FLiRTI clearly depends on the coefficient function, a single stage fit using 1st order b-splines and the Fused LASSO appears to always perform pretty well.

Table 2.6: Error rates for Z coefficient function with measurement error

γ	Fit	MISE1	MISE2	Type I errror	Type II error
0.01	Fused Lasso	0.182	0.074	0.477	0.011
	2nd Stage Ramp	0.492	0.079	0.099	0.032
	FLiRTI 1	1.363	0.106	0.854	0.074
	FLiRTI2	0.909	0.108	0.559	0.045
0.025	Fused Lasso	0.209	0.090	0.422	0.014
	2nd Stage Ramp	0.518	0.096	0.145	0.035
	FLiRTI 1	1.419	0.130	0.791	0.072
	FLiRTI2	0.929	0.126	0.555	0.045

For this simulation, the measurement error included was very small. (The standard deviation of γ was $0.01, 0.025$ compared to a standard deviation of 4 for the basis function coefficients used to generate the curves.) Since the error was small, the results were still reasonable, but we can see from even the small difference in these two levels of γ that increases in measurement error lead to poorer estimates of $\beta(t)$. For any level of γ much higher than what was described here, the curves would need to be calibrated to give unbiased estimates for $\beta(t)$.

2.6 Discussion

Sparse regression for functional data is a developing topic in the literature. As the volume of data available continues to grow, it becomes critical to be able to sort through it effectively and identify the most important information. The method we propose above is unique in that it can be based off any initial fit. The better the initial fit is, the better the final fit will be. Additionally, the second stage fit can improve the first stage fit in terms of $MISE_1$ and $MISE_2$ in addition to producing sparse fits. In this way, our method acts not only to identify sparse regions on the domain of the predictor but also to give a "second chance" at finding a good fit.

As useful as that is, one thing we saw in the simulations was that often times the single stage Fused LASSO with 1st order b-spline basis did as good of a job or even better than the competitors, including FLiRTI and the two-stage estimates. The next logical step then might be to use higher-order b-splines with the Fused LASSO. While higher-order splines overlap, if a sufficiently large basis is used, large regions of sparsity should still result.

We briefly discussed data where each curve is not fully observed and gave a few simulation results that indicate that our method works in such cases. However, even for those cases we assumed a relatively dense grid for sampling and very small measurement error. It is often the case in functional data analysis that data is observed very sparsely with error and sampled irregularly. Many methods in the literature address functional regression for this type of data, including a nonlinear approach discussed in this dissertation. A logical extension of this sparse fitting method would be to apply it in concert with these methods from the literature. To our knowledge, this would be the first attempt at sparse functional regression for sparse and irregularly sampled data.

Chapter 3

Classification for Functional Data

3.1 Introduction

It is becoming increasingly common to treat longitudinal data as sparse and irregularly sampled observations from an underlying function or curve. This so-called functional data has motivated numerous new techniques that leverage its unique characteristics to fit more accurate models. Functional Data Analysis (FDA) refers to techniques specially designed for the analysis of random trajectories. A general overview of FDA was given by Ramsay and Silverman (2005).

One important problem in the functional data literature is classification, which arises when we observe a categorical response along with one or more functional predictor variables. Let $X_i(t)$ be the observed curve of the ith experimental unit, where $t \in \mathcal{T}$ and $\mathcal{T} \subset R$ is a time interval. Let Y_i denote the class label, taking one value from $\{1, \ldots, K\}, K \geq 2$. The main task is to define a mapping $f : X(t) \rightarrow Y$, which can be used to make a future prediction.

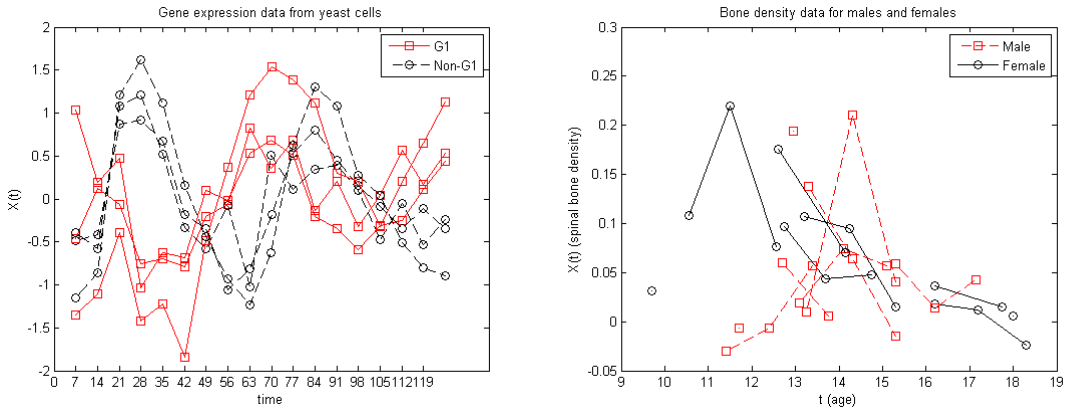


Figure 3.1: Example curves from gene expression and spinal bone mineral density data sets

Functional data classification has many real applications. Spellman et al. (1998) study temporal gene expression data from yeast cells in order to identify which genes produce varying levels of RNA over the course of the cell cycle. The authors are particularly interested in genes involved in the G1 phase of the cell cycle. Genes expression levels are observed on a fixed grid of 18 time points, and a traditional approach is used to identify which of these genes are cell cycle regulated (that is, their expression varies greatly depending on the phase of the cell cycle). A second application comes from Bachrach et al. (1999), who observe spinal bone mineral densities for males and females from ages 9 to 25 years old. Samples were taken at irregular intervals and for many of the subjects only a few observations were made. (In some cases, as few as only one or two observations were recorded.) From Figure 3.1, it is obvious that males and females have very different growth patterns. Also, the spinal bone mineral density data is quite different from the gene expression data, as the latter has been regularly sampled while the former is irregularly sampled, and some curves are observed at very few time points. The goal of this research is to develop a unified classification approach that can be used for both types of data, remains simple in its application, and produces results as good or better than existing methodology.

Classification for functional data is different in many ways from classification for multivariate data (that is, observations that are not longitudinal vectors). In particular, the functional characteristics present unique problems since functional data is often inherently infinity dimensional, and observations at near-by time points will tend to be highly correlated. Simply discretizing the curves on a grid can be suboptimal since inherent structure is not fully explored, and this approach may also lead to the curse of dimensionality if too fine a grid is used. Additionally, the observed curves may not be densely sampled or recorded at regular intervals. These features make the classification of longitudinal data more difficult.

In traditional multivariate data settings, many classification techniques have been developed and successfully applied to real problems, including linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic regression, random forests, support vector machines (SVMs), and boosting. See Hastie et al. (2009) for an overview. For functional data classification, the available tools are more limited. One of the earliest works is the functional linear discriminant analysis (FLDA) proposed by James and Hastie (2001). They first decompose the curves using a p-dimensional cubic b-spline basis and then transform the coefficient vectors into a $q < p$ dimensional space. LDA is applied to these reduced rank coefficient vectors to find the classification rule. Similarly to reduced-rank LDA for scalar data, they assume the reduced rank coefficient vectors are normally distributed and that the covariance matrices for these vectors are the same across different groups. Leng and Müller (2006) propose a method using logistic regression. First they apply functional principal components (FPC) analysis to each curve using PACE (Yao et al., 2005b). Logistic regression is used on the FPC scores to model the response. Since PACE is used to find the FPC scores, this method can be used for data that are sparsely observed and sampled at irregular intervals. Recently Li and Yu (2008) combine the LDA with support vector machines and develop a new method called functional

segment discriminant analysis. Muñoz and González (2010) use a reproducing kernel Hilbert space to find a finite-dimensional representation of each curve and classify using a support vector machine.

In this article, we propose a new simple alternative for classification of functional data. In the traditional setting, the classical method of LDA has proven to be one of the most successful methods for classification. James and Hastie (2001) extended the LDA to functional data using a basis decomposition approach. We propose an alternative framework to implement LDA for curved data through the following steps. First, we estimate the mean function for each group and subtract the mean from each observation. We refer to this as the "centering" process, since the process is analogous to subtracting the mean from scalar data to center the data's distribution at 0. This approach is different from James and Hastie (2001), which does not center the curves before performing their basis decomposition. Second, we pool the "centered" data from all of the groups and estimate a common covariance structure using FPC analysis. This step can be theoretically justified under the LDA assumption of equal variances across groups. We obtain estimates for the eigenfunctions (a smooth orthogonal basis expansion) and corresponding eigenvalues of the underlying covariance structure. This allows us to represent each curve as a mean function plus a sum of these eigenfunctions multiplied by their corresponding FPC scores, a form known as the Karhunen-Loève representation. We truncate the expansion and use the basis function coefficients to create a classification rule.

The remainder of the chapter is organized as follows. Section 2 discusses our method in detail as well as an alternative approach for a special case. Section 3 provides theoretical results for the new approach. Section 4 compares our methods with existing methods using simulated data as well as three data sets from the literature. Section 5 concludes with a discussion.

3.2 New Methodology

3.2.1 General Setup and Notations

For subject i, we observe the curve $X_i(t)$ at N_i random time points, denoted as $T_{i1}, ..., T_{iN_i}$, on some domain, $\mathcal{T}$, where N_i is a random variable independent of the other random variables. To be consistent with real applications, we allow observations to be made with measurement errors, denoted as δ_{ij}. Therefore, the observed values for the ith curve at time T_{ij} are

$$U_{ij} = X_i(T_{ij}) + \delta_{ij}, \;\; j = 1, 2, \ldots, N_i.$$

The measurement errors are assumed be independent both between and within curves with mean 0 and $var(\delta_{ij}) < \infty$. This setup has been considered by Yao et al. (2005b) and many others. For the binary response case, the class label $Y_i = 1$ if curve $X_i(t)$ belongs to Class 1 and $Y_i = 2$ if $X_i(t)$ belongs to Class 2. For the multiclass case, the class label is from $\{1, \ldots, K\}$.

For each class k, we denote the class-mean function by $\mu_k(t) = E(X_i(t)|Y_i = k)$ for $t \in \mathcal{T}$. For each curve, we denote the covariance function $V(s,t) = \text{cov}(X_i(s), X_i(t))$ for $s, t \in \mathcal{T}$. In our procedure, we assume a common $V(s,t)$ across different classes, which corresponds to the key equal-covariance assumption employed in the classical LDA. In practice, if the assumption fails, one can extend the procedure to quadratic discriminant analysis.

Let $\lambda_j, \phi_j(\cdot), j = 1, 2, \ldots$ be the eigenvalues and eigenfunctions of the autocovariance operator of $X(t)$ where $\lambda_1 \geq \lambda_2 \geq \ldots$. Then the covariance function $V(s,t)$ can be represented using

an orthogonal basis expansion, $\{\phi_j(t)\}_{j=1}^{\infty}$, i.e.,

$$V(s,t) = \sum_{j=1}^{\infty} \lambda_j \phi_j(s)\phi_j(t). \tag{3.1}$$

Correspondingly we can write each curve as

$$X_i(t) = \mu_k(t) + \sum_{l=1}^{\infty} \xi_{ij}\phi_j(s), \quad \text{if} \quad Y_i = k, \tag{3.2}$$

where ξ_{ij} are the FPC scores computed as $\xi_{ij} = \int_{\mathcal{T}}[X_i(t) - \mu_k(t)]\phi_j(t)dt, j = 1, 2, \ldots$. The FPC scores ξ_{ij} are uncorrelated with mean 0 and variance var$(\xi_{il}) = \lambda_j$ for $j = 1, 2, \ldots$. This representation is known as the Karhunen-Loève expansion.

3.2.2 New Functional Discriminant Analysis

We first illustrate the approach for the binary problem. Define the observed data as $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{n}$, where $\mathcal{D}_i = \{N_i, (T_{i1}, U_{ij}), \ldots, (T_{iN_i}, U_{iN_i})\}$ are the observed values for the ith curve. The first step is to center the data. Following Yao et al. (2005b), we apply a local polynomial smoother to $\{(T_{ij}, U_{ij}), j = 1, 2, \ldots, N_i : Y_i = k\}$ in order to obtain an estimate $\hat{\mu}_k(t)$ of class mean function $\mu_k(t)$ for each group k. A complete introduction of local polynomial modeling is given in Fan and Gijbels (1996). We take these mean functions and subtract them from the data to get de-meaned or "centered" data. Let

$$C_{ij} = U_{ij} - \widehat{\mu}_0(T_{ij})\mathbf{1}[i \in I_0] - \widehat{\mu}_1(T_{ij})\mathbf{1}[i \in I_1], \tag{3.3}$$

where $I_k = \{i : Y_i = k\}$ and $\mathbf{1}$ is an indicator function.

With the "centered" data, we estimate the eigenfunctions $\phi_l(t)$ and eigenvalues λ_j using functional principal components analysis through conditional expectation (PACE) (Yao et al., 2005b). PACE is an ideal method for this problem, since it provides estimates for full curves even when the curve is observed at only a few time points, and software is readily available for its implementation in Matlab. As we have already de-meaned the data in the previous centering step, we skip the step of estimating mean function while applying PACE. Since the groups are assumed to share a covariance structure, we pool the centered data together before using PACE to estimate the eigenfunctions, eigvenvalues, and FPC scores. We denote our estimate of the covariance function as

$$\widehat{V}(s,t) = \sum_{l=1}^{J} \widehat{\lambda}_l \widehat{\phi}_l(s) \widehat{\phi}_l(t) \tag{3.4}$$

for some $J > 0$, where $\widehat{\lambda}_l$ is the estimate of the lth eigenvalue, and $\widehat{\phi}_l(t)$ the estimate of the corresponding eigenfunction.

When the data being smoothed over has a common mean function as described in Yao et al. (2005b), estimates of the mean function, eigenfunction, and eigenvalues have been shown to be consistent (Yao et al., 2005b). Although our application involves data from multiple groups, each with its own mean function, showing the consistency of these estimates is only slightly different.

Once we have estimated eigenvalues and eigenfunctions, we can obtain estimates $\hat{\xi}_{il}$ of the FPC scores ξ_{il} for each curve by applying conditional expectation as done in PACE. Correspondingly, an estimate of the full curve for each subject is given by

$$\widehat{X}_i(t) = \widehat{\mu}_k(t) + \sum_{l=1}^{J} \widehat{\xi}_{il}\widehat{\phi}_l(t), \tag{3.5}$$

where $Y_i = k$.

We can now define a classification rule based on the estimates presented above. Suppose we have a random number (N^*) of sparse and irregular observations, denoted by $(T_j^*, U_j^*), j = 1, 2, \ldots, N^*$, from a new curve, $X^*(t)$, whose class label is unknown. We classify based on our estimates of the mean functions from the different classes and the estimated eigenvalues and eigenfunctions. The first step is to estimate FPC scores using conditional expectation for each possible group by pretending that $X^*(t)$ is from this group. That is, we find FPC score estimates $\hat{\xi}_{1k}^*, \hat{\xi}_{2k}^*, \ldots, \hat{\xi}_{Jk}^*$ for each group k by first "centering" $(T_j^*, U_j^*), j = 1, 2, \ldots, N^*$, using the estimated mean function for group k and estimating the expectation of the FPC scores conditional on observing $T_1^*, \ldots, T_{N^*}^*$. This is the PACE estimate of the FPC scores. These estimated FPC scores can be thought of as estimates of the true FPC scores for this new observation, conditional on the observation belonging to Class k. Since FPC scores are assumed to be normally distributed with mean zero and variance equal to the corresponding eigenvalue, we can write the likelihood that our new curve, $X^*(t)$ belongs to Class k as

$$L_k = \Pi_{l=1}^{J} f_{\widehat{\lambda}_l}(\widehat{\xi}_{kl}^*), \tag{3.6}$$

$f_{\widehat{\lambda}_l}$ is the pdf for a *Normal*$(0, \widehat{\lambda}_l)$. When k takes only two values, $1, 2$, can make class membership predictions for $X^*(t)$ based on the ratio of the two likelihoods

$$\frac{L_1}{L_2} = \frac{Exp\left\{-.5\sum_{l=1}^{J}\frac{\widehat{\xi}_{1l}^{*2}}{\widehat{\lambda}_l}\right\}}{\left\{Exp-.5\sum_{l=1}^{J}\frac{\widehat{\xi}_{2l}^{*2}}{\widehat{\lambda}_l}\right\}}. \tag{3.7}$$

Then if $\frac{L_1}{L_2} > 1$, our class membership prediction is Class 1. Otherwise, we predict it as a member of Class 2. This rule easily generalizes to the mutliclass case using the argmax rule

$$\arg\max_k \eta_k(X(t)), \tag{3.8}$$

where $\eta_k(X(t)) = Exp\{-.5\sum_{j=1}^{J}\frac{\widehat{\xi}_{kj}^{*2}}{\widehat{\lambda}_j}\}$. When $K > 2$, we can find the value for each L_k and choose the class with the highest associated likelihood.

3.2.3 Uncentered Approach

In the method proposed above, we implicity assume that the difference of the mean functions, $\mu_k(t)$, is in the space spanned by eigenfunctions, $\{\phi_l(t)\}_{l=1}^{J}$, as otherwise the mean function difference orthogonal to these eigenfunctions is filtered by the truncated covariance operator $\sum_{l=1}^{J}\lambda_l\phi_l(s)\phi_l(t)$. However functional data is inherently infinitely dimensional, and consequently it is possible that the mean function difference is not in the space spanned by the first J eigenfunctions. This motivates us to propose the following uncentered approach.

As mentioned earlier, the success of the discriminant analysis method proposed in the previous section requires that $\mu_1(t) - \mu_2(t) = \sum_{l=1}^{J}\mu_l\phi_l(t)$ for some coefficients, μ_l. As $J \to \infty$, this issue will not arise theoretically. But in practice since the choice of J is based only on data and estimated covariance structure, it is possible that we may choose J too small to capture

the difference between mean functions. Suppose μ_1 is orthogonal to $\phi_l(t)$ for all $l = 1, \ldots, J$. Using the method described above to classify a new curve, we estimate FPC scores after using each mean function to center the data. Therefore, when calculating ξ^*_{1l}, we would need to project $X^*(t) - \mu_1(t)$ onto $\{\phi_l(t)\}_{l=1}^J$. Since $\mu_1(t)$ is orthogonal to $\{\phi_l(t)\}_{l=1}^J$, the resulting estimates will be unstable.

To illustrate this in details, we consider the two-class case again and use the following notations. Let $X^1(t)$ denote a curve from Class 1 with mean function, $\mu_1(t)$, $X^2(t)$ denote a curve from Class 2 with $\mu_2(t)$. The overall mean function of the two groups is a weighted average of these mean functions weighted by group proportions. Thus we write $\bar{\mu}(t) = \tau_1\mu_1(t) + \tau_2\mu_2(t)$ with $\tau_1 + \tau_2 = 1$.

For curves from Classes 1 and 2, we can then write

$$X_i^1(t) - \bar{\mu}(t) = \tau_1(\mu_1(t) - \mu_2(t)) + \sum_{l=1}^{J} \xi_{il}\phi_l(t) \tag{3.9}$$

$$X_i^2(t) - \bar{\mu}(t) = \tau_2(\mu_2(t) - \mu_1(t)) + \sum_{l=1}^{J} \xi_{il}\phi_l(t). \tag{3.10}$$

Consequently a general curve $X(t)$ can be represented as

$$X(t) = \bar{\mu}(t) + \sum_{l=1}^{J} \xi_l\phi_l(t) + \xi(\mu_1(t) - \mu_2(t)), \tag{3.11}$$

where $\xi = \tau_1$ if $X(t)$ belongs to group 1 and τ_2 otherwise. Denote $\phi(t) = \mu_1(t) - \mu_2(t) - \sum_{l=1}^{J} \phi_l(t)\xi_l^d$, with $\xi_l^d = \int_{\mathcal{T}}(\mu_1(s) - \mu_2(s))\phi_l(s)ds$ to be the part of $\mu_1(t) - \mu_2(t)$ that is

orthogonal to $\phi_1(t), \phi_2(t), \ldots, \phi_J(t)$. Then we have

$$X(t) = \bar{\mu}(t) + \sum_{l=1}^{J} (\xi \xi_l^d + \xi_l)\phi_l(t) + \xi\phi(t), \tag{3.12}$$

which leads to the Karhunen-Loève expansion of the combined data. Here the last term may disappear if the mean function difference is in the space spanned by the eigenfunctions. Note that the classification power is complemently determined by ξ as it takes different values for different groups. Now it becomes clear why the success of the new discriminant analysis method proposed in the previous section depends heavily on whether the mean function differences fall in the space spanned by the eigenfunctions. Note that $\xi_l^d = 0$ for $l = 1, 2, \ldots, J$ if the mean function difference is orthogonal to these eigenfunctions. Correspondingly the FPC scores $\xi\xi_l^d + \xi_l$ do not differ between these two groups and we need to recruit "additional" eigenfunction "$\phi_{J+1}(t) = \phi(t)$" to discriminant these two classes. To perform classification for this type of data, it is sufficient to perform LDA on the FPC scores estimated by applying the PACE to the overall data, which should include the estimated FPC score corresponding to the "additional" eigenfunction "$\phi_{J+1}(t)$".

To address aforementioned difficulty in situations with mean function difference not in the space spanned by the eigenfunctions, we propose an alternative method where the "centering" step is skipped. An overall mean function $\bar{\mu}(t)$ is used instead and we apply PACE to the data $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n$ from all groups to estimate FPC scores. We then apply traditional LDA to the FPC score estimates, giving us a classification rule.

Finally, note that while the "centered" method, introduced first, may have trouble with the type of data described here, the "uncentered" approach described here may be used even when the

mean function difference is in the space spanned by the eigenfunctions as shown in (Eq. 3.12). In this sense, the "uncentered" approach is more robust.

3.2.4 Summary of Algorithms

We summarize the two methods described above as follows.

Centered method:

1. Estimate the mean function for each group using local polynomial smoothing.
2. De-mean observations from each curve by subtracting the corresponding estimated group mean function.
3. Apply PACE to the centered data to estimate the eigenvalues and eigenfunctions of the common covariance operation.
4. For observations on a new curve, estimate the corresponding FPC scores for each class, $k = 1, \ldots, K$ by pretending it comes from this group and subtracting off the corresponding estimated group mean function.
5. Choose k such that the estimated FPC scores maximize the likelihood.

Uncentered method:

1. Without centering the data, combine the data from all different groups and apply PACE to estimate the FPC scores.
2. Apply LDA to the estimated FPC scores to find the classification rule.

3. For new observations from a new curve, estimate the corresponding FPC scores and perform classification according to the rule above.

3.3 Consistency and Computation Issues

3.3.1 Consistency of estimated FPC scores

In this section, we discuss the consistency of the proposed new discriminant analysis.

For the classification of functional data, there are two layers of Bayes error. The first level is that we assume the functional curve to be fully observable and the covariance function $V(s,t)$ and its eigen decomposition $V(s,t) = \sum_{=1}^{\infty} \lambda_l \phi_l(s)\phi_l(t)$ to be known. For a new curve $X^*(t)$, its loglikelihoods to belong to Classes 1 and 2 are given by

$$-\int_s \int_t (X^*(t) - \mu_1(t))(\sum_{l=1}^{\infty} \frac{1}{\lambda_l}\phi_l(t)\phi_l(s))(X^*(s) - \mu_1(s))dsdt$$

and

$$-\int_s \int_t (X^*(t) - \mu_2(t))(\sum_{l=1}^{\infty} \frac{1}{\lambda_l}\phi_l(t)\phi_l(s))(X^*(s) - \mu_2(s))dsdt,$$

respectively, up to a same constant. Denote $\xi^*_{1l} = \int_t (X^*(t) - \mu_1(t))\phi_l(t)dt$ and $\xi^*_{2l} = \int_t (X^*(t) - \mu_2(t))\phi_l(t)dt$. The corresponding Bayes rule is to choose $\widehat{Y}^* = 1$ if

$$-\sum_{l=1}^{\infty} (\xi^*_{1l})^2/\lambda_j > -\sum_{l=1}^{\infty} (\xi^*_{2l})^2/\lambda_l$$

and choose $\widehat{Y}^* = 2$ otherwise. The corresponding Bayes error is given by

$$E(Y \neq I(-\sum_{l=1}^{\infty}(\xi_{1l}^*)^2/\lambda_l > -\sum_{l=1}^{\infty}(\xi_{2l}^*)^2/\lambda_l)).$$

Sparse and irregularly observed data introduce another level of error. For the new curve, $X^*(t)$, we observe only $(T_j^*, U_j^*), j = 1, 2, \ldots, N^*$. Consequently we do not have the complete information about the FPC scores ξ_{1l}^* and ξ_{2l}^*, $l = 1, 2, \ldots$. The best estimate of these FPC scores are their conditional expectations $E(\xi_{1l}^*|\{(T_j^*, U_j^*)\}_{j=1}^{N^*})$ and $E(\xi_{2j}^*|\{(T_j^*, U_j^*)\}_{j=1}^{N^*}))$, where $j = 1, 2, \ldots$. In this case, the corresponding Bayes rule is to choose $\widehat{Y}^* = 1$ if

$$-\sum_{j=1}^{\infty}(E(\xi_{1l}^*|\{(T_j^*, U_j^*)\}_{j=1}^{N^*}))^2/\lambda_j \geq -\sum_{j=1}^{\infty}(E(\xi_{2l}^*|\{(T_j^*, U_j^*)\}_{j=1}^{N^*})))^2/\lambda_j \tag{3.13}$$

and 2 otherwise.

According to Yao et al. (2005b), the estimates of mean functions $\mu_1(t)$ and $\mu_2(t)$, covariance function $V(s,t)$, eigenvalue λ_j, eigenfunction $\phi_j(t)$, and conditional expectation of FPC scores $E(\xi_{1j}^*|(T_1^*, U_1^*), \ldots, (T_{N^*}^*, U_{N^*}^*))$ and $E(\xi_{2j}^*|(T_1^*, U_1^*), \ldots, (T_{N^*}^*, U_{N^*}^*))$ are consistent. Consequently our new discriminant analysis is consistent as long as we allow the number (J) of included FPC scores to diverge to infinity as n approaches infinity.

3.3.2 Consistency of classification rule

One of the most compelling properties of LDA is that the classification rule it defines is equivalent to the Bayes rule. Our method has this property, as well. If we assume that the vector of

functional principal component scores, $\boldsymbol{\xi_i}$, has some underlying distribution f_k when $Y_i = k$, then from Bayes' formula, we have

$$P(Y_i = K|\boldsymbol{\xi_i}) = \frac{f_k(\boldsymbol{\xi_i})\pi_i}{\sum_{m=1}^{K} f_m(\boldsymbol{\xi_i})\pi_m}, \tag{3.14}$$

where π_m is $P(Y_i = m)$. The classifier that sets $\widehat{Y}_i = k$ where k maximizes Eq. 3.14 is known as the Bayes rule. If the ξ_{ij} are Gaussian with mean 0 and variance λ_J, then

$$\frac{L_0}{L_1} = \frac{Exp - .5\sum_{l=1}^{J} \frac{\xi_{0l}^{*2}}{\lambda_l}}{Exp - .5\sum_{l=1}^{J} \frac{\xi_{1l}^{*2}}{\lambda_l}} \tag{3.15}$$

is the Bayes rule for the two-class case, provided that $E(\xi_{kl}^*) = \xi_{kl}$. However, PACE estimates the conditional expectation of the FPC scores, so in fact we have $\widehat{\xi}^* = \widehat{E}[\xi_{kl}^*|\boldsymbol{T}_i]$. Thus, as long as we use PACE to estimate the FPC scores, we cannot claim that our classifier converges to the Bayes rule. However, for data sampled over a fine grid, it is possible to estimate the unconditional scores for the FPC functions. In such a case, the "centered" method would converge to the Bayes rule.

3.3.3 Computation

One advantage our method has over others is the relative ease of computation. Once the FPC scores have been estimated for each group, the computation time is trivial. This means the bulk of the cost is paid due to PACE, which is relatively efficient. Computation time comparisons are included in the simulations reported in the following section.

3.4 Simulation

We generate data using three eigenfunctions,

$$\phi_1(t) = -\sqrt{\frac{1}{5}} cos\left(\frac{\pi t}{5}\right), \tag{3.16}$$

$$\phi_2(t) = \sqrt{\frac{1}{5}} sin\left(\frac{\pi t}{5}\right), \tag{3.17}$$

$$\phi_3(t) = -\sqrt{\frac{1}{5}}\left(\frac{\pi 2t}{5}\right), \tag{3.18}$$

defined over the domain $\mathcal{T} = (0, 10)$. Each curve, $X_i(t) = \mu_k(t) + \sum_{l=1}^{3} \xi_{il}\phi_l(t)$, where $\xi_{il} \sim N(0, \lambda_l)$. For $i = 1, \ldots, n$, we observe $U_i(T_{ij}) = X_i(T_{ij}) + \epsilon_{ij}$, where $j = 1, \ldots, N_i$, the T_{ij} are *idd* uniformly on $\mathcal{T}$, $\epsilon \sim N(0, \sigma^2)$, and $N_i \sim UD(5, N)$, where UD is the discrete uniform distribution, $\sigma = 0.5$, and n and N are simulation parameter. An additional simulation parameter, c, determines the separation between the two mean functions.

The number of eigenfunctions used for classification, J, is a tuning parameter, and we choose it using five-fold cross-validation, where J is chosen to minimize the classification error.

For comparison with the methods described above, we consider two alternative methods from the literature, Functional Linear Discriminant Analysis (FLDA) (James et al., 2000) and logistic regression (Leng and Müller, 2006).

3.4.1 Case 1

First, we examine the case where $\mu_0(t)$ and $\mu_1(t)$ are in the space spanned by $\{\phi_l(t)\}_{l=1}^{3}$. Let

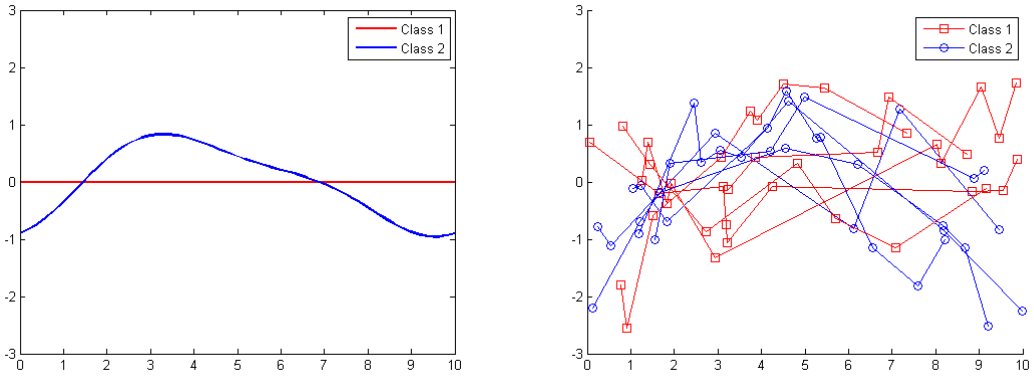


Figure 3.2: Case 1 mean functions and simulated data points

$$\mu_0(t) = 0, \tag{3.19}$$

$$\mu_1(t) = \frac{1}{c}\sum_{i=1}^{3}\phi_1(t). \tag{3.20}$$

The simulation parameter, c, determines how far apart the two groups are, with larger values of c corresponding to closer groups and hence a more difficult problem. We let $n = 50, 100$, (with an equal number of observations in the two groups) $N = 10, 20$ and $c = 1, 2$. The results are summarized in Table 3.1. The median of the standard deviations in each column are given across the bottom row of the table. A plot of the two mean functions along with observed curves is given in Figure 3.2.

Both of our methods perform well for this data relative to the alternatives. In particular, when $n = 50$, our method shows substantial gains relative to FLDA. Performance is also superior when compared with logistic regression, although the differences are not as large. Additionally, we can see that FLDA takes much longer than the other methods, and the "centered" method

Table 3.1: Classification errors for Case 1 (computation time in minutes)

			Error Rates					Computation Time			
N	n_k	c	Cen	Uncen	Log	FLDA	Bayes	Cen	Uncen	Log	FLDA
10	50	1	0.170	0.171	0.172	0.180	0.123	0.892	1.498	0.967	3.684
10	50	1.5	0.249	0.249	0.248	0 .250	0.193	0.888	1.540	0.998	3.765
10	50	2	0.334	0.331	0.331	0.323	0.286	0.898	1.519	1.049	3.741
10	50	2.5	0.362	0.365	0.369	0.377	0.327	0.897	1.535	1.048	3.672
10	100	1	0.179	0.176	0.174	0.177	0.131	1.713	3.536	1.952	4.012
10	100	1.5	0.265	0.264	0.264	0.278	0.233	1.710	3.614	2.034	4.159
10	100	2	0.324	0.326	0.326	0.327	0.305	1.753	3.609	2.055	4.195
10	100	2.5	0.350	0.347	0.350	0.357	0.320	1.751	3.560	2.095	4.256
20	50	1	0.170	0.170	0.172	0.185	0.142	2.269	3.041	2.385	12.778
20	50	1.5	0.261	0.257	0.260	0.268	0.232	2.304	3.137	2.512	13.135
20	50	2	0.339	0.342	0.343	0.353	0.312	2.350	3.232	2.544	13.011
20	50	2.5	0.369	0.371	0.370	0.373	0.339	2.330	3.159	2.530	13.395
20	100	1	0.152	0.148	0.149	0.163	0.119	5.045	7.671	5.421	16.880
20	100	1.5	0.246	0.245	0.243	0.256	0.212	5.199	7.847	5.586	16.960
20	100	2	0.316	0.317	0.318	0.335	0.304	4.752	7.760	5.256	15.029
20	100	2.5	0.347	0.347	0.348	0.365	0.332	4.684	7.170	5.188	15.252
	SD		0.016	0.014	0.014	0.016					

is the most computationally efficient of the group.

3.4.2 Case 2

Next, we look at a situation where the mean functions are not in the space spanned by the eigenfunctions. Now, let

$$\mu_0(t) = 0, \tag{3.21}$$

$$\mu_1(t) = 2. \tag{3.22}$$

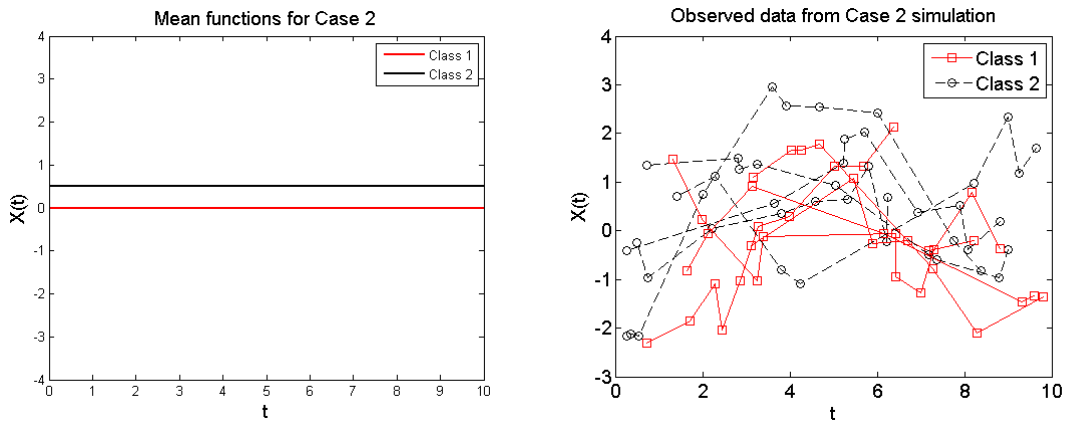


Figure 3.3: Case 2 mean functions and simulated data points

In this situation, we would expect to see the "uncentered" method perform substantially better than the "centered" method. We can see the true mean functions along with observed curves for Case 2 in Figure 3.3.

The results from Case 2 are interesting, and we see a few clear trends as well as some ambiguous ones. First, the "centered" method performs poorly in this case relative to all other methods. This was expected, as the example was constructed to be a special case where the basis functions used to generate the data were orthogonal to the mean functions from the two classes. However, the "uncentered" method is competitive with the alternatives. When relatively little information is available ($n_k = 50$ and $N = 10$ small), FLDA is slightly better than the competition, but when more information is available ($n_k = 100$ and $N = 20$), both the "uncentered" method and logistic regression surpass FLDA. This is most pronounced when the classification problem is more difficult ($c = 2, 2.5$). Additionally, we see that when c is large, logistic regression and FLDA have very similar classification error rates, but when c is small, our method is superior to logistic regression. Overall, this indicates that for the sparsest cases where the mean function is orthogonal to the basis selected to model the data, FLDA may be a

Table 3.2: Classification errors for Case 2 (computation time in minutes)

			Error Rates					Computation Time			
N	n_k	c	Cen	Uncen	Log	FLDA	Bayes	Cen	Uncen	Log	FLDA
10	50	1	0.117	0.048	0.057	0.049	0.234	1.503	1.364	0.949	5.424
10	50	1.5	0.210	0.113	0.121	0.108	0.296	1.495	1.405	0.985	5.5
10	50	2	0.316	0.206	0.209	0.199	0.353	1.517	1.472	1.027	4.716
10	50	2.5	0.355	0.260	0.258	0.238	0.393	1.514	1.496	1.054	4.591
10	100	1	0.106	0.036	0.041	0.035	0.219	2.999	3.323	1.929	11.376
10	100	1.5	0.217	0.094	0.096	0.101	0.304	2.864	3.213	1.928	10.090
10	100	2	0.273	0.173	0.173	0.177	0.343	2.812	3.165	1.907	10.244
10	100	2.5	0.340	0.236	0.237	0.229	0.376	2.	3.276	2.004	9.797
20	50	1	0.090	0.013	0.023	0.012	0.287	3.865	2.920	2.480	35.560
20	50	1.5	0.190	0.055	0.066	0.067	0.341	4.101	3.300	2.785	36.270
20	50	2	0.274	0.106	0.112	0.117	0.375	4.356	3.610	2.830	30.008
20	50	2.5	0.350	0.165	0.167	0.173	0.409	4.443	3.696	2.929	27.243
20	100	1	0.079	0.015	0.019	0.016	0.257	7.802	6.557	4.853	32.237
20	100	1.5	0.169	0.050	0.055	0.058	0.317	7.716	6.659	4.860	31.798
20	100	2	0.260	0.102	0.105	0.122	0.380	7.495	6.431	4.674	30.563
20	100	2.5	0.320	0.146	0.146	0.168	0.445	7.879	7.040	5.058	29.482
	SD		0.064	0.015	0.017	0.007					

slightly better choice than the "centered" method. However, when there is more data available, the "uncentered" method is clearly as good as or better than the alternatives.

3.4.3 Multiclass simulations: $K > 2$

We also present results from two multiclass simulations. We use data from three classes for the first simulation and four classes for the second. In both cases, the the mean functions are in the span of $\{\phi_l(t)\}_{l=1}^3$. For the $K = 3$ case, let

$$\mu_1(t) = \frac{1}{c}\phi_1(t), \tag{3.23}$$

$$\mu_2(t) = \frac{1}{c}\phi_2(t), \tag{3.24}$$

$$\mu_3(t) = \frac{1}{c}\phi_3(t), \tag{3.25}$$

where $\{\phi_l(t)\}_{l=1}^3$ is the same as from the two-class simulations. The mean functions for this simulation as well as the other mutliclass simulations below can be seen in Figure 3.4. We once again vary the sparsity of the data, N, the number of observations in each group, n_k, and the separation of the groups, c. The classification error rates based on a test data set of $n_{test} = 1000$ are summarized below for the centered and uncentered methods as well as FLDA. Logistic regression was not included in these simulations.

While the "centered" method lags behind, FLDA and the "uncentered" method both show good performance for this three-class simulation, with the "uncentered" method showing superior results at every stage. One thing to note is that the difference between the error rates for the "uncentered" method and FLDA increase with c, so the advantage for using this method over FLDA increases as the problem becomes more difficult.

Next, we look at the case where $K = 4$. Once again the mean functions are equally spaced using the points of a regular tetrahedron. The mean functions are given as

Table 3.3: Classification errors 3-class simulation (computation times in minutes)

			Error Rates				Computation Time		
N	n_k	c	Cen	Uncen	FLDA	Bayes	Cen	Uncen	FLDA
10	40	1	0.082	0.039	0.054	0.013	1.904	1.734	6.287
10	40	1.5	0.207	0.141	0.159	0.082	1.940	1.749	6.609
10	40	2	0.320	0.233	0.247	0.166	1.958	1.798	6.736
10	40	2.5	0.408	0.326	0.337	0.257	1.999	1.768	6.757
10	80	1	0.055	0.042	0.049	0.018	3.311	3.821	9.076
10	80	1.5	0.163	0.125	0.138	0.076	3.353	3.751	7.531
10	80	2	0.273	0.228	0.246	0.174	3.346	3.765	8.799
10	80	2.5	0.354	0.312	0.332	0.240	3.319	3.748	9.129
20	40	1	0.067	0.022	0.032	0.007	4.544	3.432	35.943
20	40	1.5	0.192	0.121	0.142	0.084	4.658	3.593	36.403
20	40	2	0.311	0.192	0.223	0.159	4.705	3.677	36.575
20	40	2.5	0.395	0.279	0.308	0.238	4.792	3.761	37.605
20	80	1	0.044	0.027	0.039	0.017	8.465	7.358	31.335
20	80	1.5	0.138	0.098	0.114	0.071	8.550	7.343	33.297
20	80	2	0.252	0.198	0.230	0.162	9.063	7.959	28.028
20	80	2.5	0.318	0.258	0.296	0.234	8.959	8.021	27.348
	SD		0.041	0.009	0.011				

$$\mu_1(t) = \frac{1}{c}[3 \quad 3 \quad 3]\boldsymbol{\phi}(t), \tag{3.26}$$

$$\mu_2(t) = \frac{1}{c}[3 \quad -3 \quad -3]\boldsymbol{\phi}(t), \tag{3.27}$$

$$\mu_3(t) = \frac{1}{c}[-3 \quad 3 \quad -3]\boldsymbol{\phi}(t), \tag{3.28}$$

$$\mu_4(t) = \frac{1}{c}[-3 \quad -3 \quad 3]\boldsymbol{\phi}(t). \tag{3.29}$$

Classification error rates are summarized below.

Similar to the three-class case, the uncentered method clearly outperforms FLDA. However,

Table 3.4: Classification errors for 4-class simulation (computation times in minutes)

			Error Rates				Computation Time		
N	n_k	c	Cen	Uncen	FLDA	Bayes	Cen	Uncen	FLDA
10	30	1	0.071	0.027	0.141	0.004	1.932	1.606	6.684
10	30	1.5	0.202	0.112	0.249	0.060	1.937	1.639	6.772
10	30	2	0.339	0.226	0.342	0.153	1.930	1.633	6.935
10	30	2.5	0.437	0.313	0.422	0.240	1.936	1.624	6.852
10	60	1	0.049	0.035	0.145	0.008	3.778	3.896	11.912
10	60	1.5	0.151	0.108	0.226	0.064	3.674	3.789	11.100
10	60	2	0.263	0.201	0.332	0.126	3.846	3.989	12.310
10	60	2.5	0.371	0.302	0.410	0.220	3.740	3.852	11.628
20	30	1	0.070	0.015	0.136	0.004	4.799	3.422	38.615
20	30	1.5	0.218	0.101	0.252	0.055	4.886	3.581	37.273
20	30	2	0.334	0.207	0.338	0.169	5.025	3.778	38.273
20	30	2.5	0.440	0.298	0.411	0.218	5.005	3.795	37.403
20	60	1	0.040	0.014	0.131	0.005	8.790	7.646	43.328
20	60	1.5	0.146	0.082	0.239	0.058	8.898	7.851	45.493
20	60	2	0.252	0.176	0.312	0.145	9.033	7.986	45.832
20	60	2.5	0.364	0.277	0.401	0.226	9.077	7.906	47.128
	SD		0.046	0.008	0.037				

unlike the three-class case, the "centered" method also outperforms FLDA for the cases where c is small. When the mean functions for the three classes are closer together (that is, c is larger), FLDA overtakes the uncentered method when the sample size is small. It is interesting to note that we don't see this trend when $c = 60$. In fact, the centered method is much closer to the uncentered method for this case and has a statistically significant difference from FLDA in this case as well.

Finally, we look at the a five-class case. The mean functions are the same as the four-class case with the addition of a class centered at

$$\mu_5(t) = \frac{1}{c}[-3 \qquad -3 \qquad -3]\boldsymbol{\phi}(t). \tag{3.30}$$

Results are given in Table 3.5.

Table 3.5: Classification errors for 5-class simulation (computation time in minutes)

			Error Rates				Computation Time		
N	n_k	c	Cen	Uncen	FLDA	Bayes	Cen	Uncen	FLDA
10	25	1	0.164	0.074	0.228	0.038	2.248	1.936	8.964
10	25	1.5	0.369	0.237	0.360	0.184	2.283	1.953	9.256
10	25	2	0.476	0.330	0.444	0.269	2.296	1.995	9.167
10	25	2.5	0.558	0.419	0.518	0.376	2.325	1.993	9.265
10	50	1	0.102	0.073	0.222	0.038	3.960	4.044	7.626
10	50	1.5	0.267	0.195	0.341	0.153	3.965	4.010	7.755
10	50	2	0.381	0.293	0.424	0.273	3.974	4.003	7.578
10	50	2.5	0.493	0.412	0.502	0.353	3.964	4.042	7.501
20	25	1	0.155	0.059	0.202	0.041	5.073	3.648	34.623
20	25	1.5	0.330	0.169	0.321	0.136	5.305	3.898	36.262
20	25	2	0.473	0.298	0.424	0.278	5.120	3.664	35.823
20	25	2.5	0.563	0.386	0.506	0.399	5.123	3.662	36.520
20	50	1	0.106	0.056	0.210	0.045	9.652	8.389	50.325
20	50	1.5	0.253	0.168	0.333	0.139	9.384	7.973	49.562
20	50	2	0.395	0.287	0.426	0.269	9.393	7.998	51.645
20	50	2.5	0.490	0.393	0.507	0.374	9.315	7.803	46.763
	SD		0.054	0.010	0.025				

These results mirror those of the 4-class simulation in that the "uncentered" method clearly outperforms the competitors. We also see the trend of the "centered" method beating FLDA for small sample sizes while FLDA proves superior for low sample sizes with c large.

3.5 Real data

Recall the data sets from the literature discussed in the introduction. Spinal bone mineral density for males and females was observed at irregular time intervals with as few as $N_i = 2$

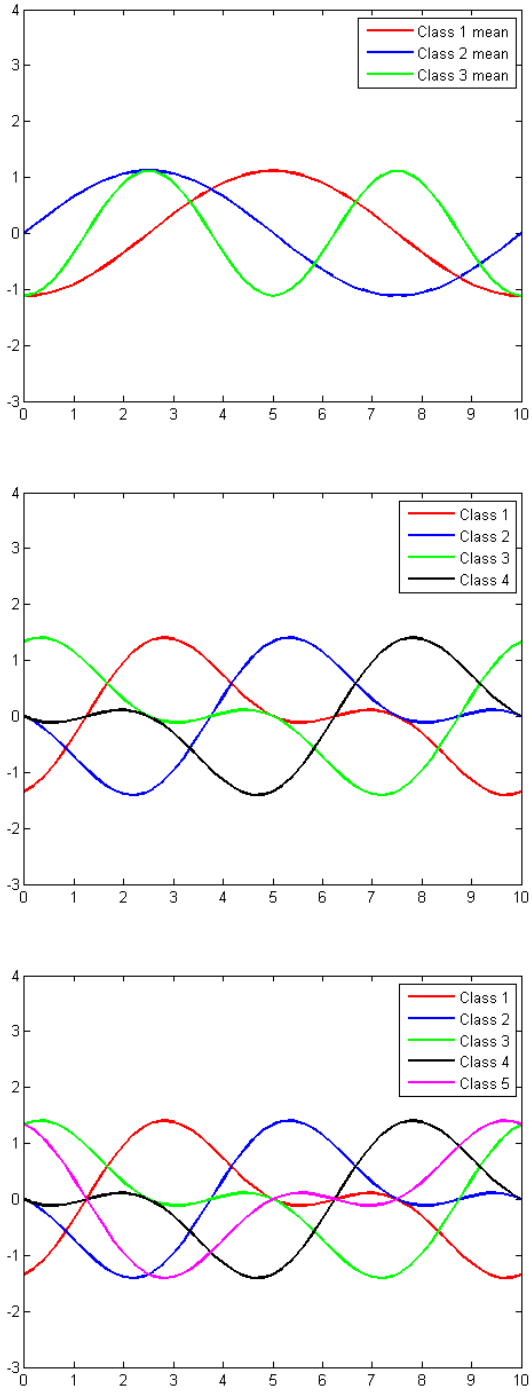


Figure 3.4: Multiclass mean functions and simulated data points

observations for some individuals. Gene expression data from yeast cells was observed at regular time intervals. Additionally, we look at the Berkeley Growth Data, where 39 male and 54 female subects were observed at 31 fixed time points from ages 1 to 18. (Ramsay and Silverman, 2005) For each data set, we applied our two methods plus FLDA and logistic regression. For the growth curves and spinal bone mineral density data sets, we classified based on gender. For the gene expression data, we classified genes based on their role in the cell cycle (G1 or non-G1). Table 3.6 below summarizes the results.

Table 3.6: Classification errors for "real world" data sets

Data Set	Centered	Uncentered	Logistic	FLDA
Growth Curves	0.1166	0.0938	0.1361	0.1695
Genes	0.1748	0.1858	0.1538	0.1978
Bone Density	0.4049	0.3269	0.3500	0.3923

The growth curves had the lowest average error rate, with the two methods proposed above showing the best performance. The "uncentered" method also performed the best for the bone density data. Logistic regression had the lowest error rate for the genetics data, which was the article where the logistic regression method was proposed. FLDA shows the worst performance for every method but the bone density data, where it beat only our"centered" method. Figure 3.5 shows the estimated curves for the gene expression data along with the original observed curves and estimated FPC functions.

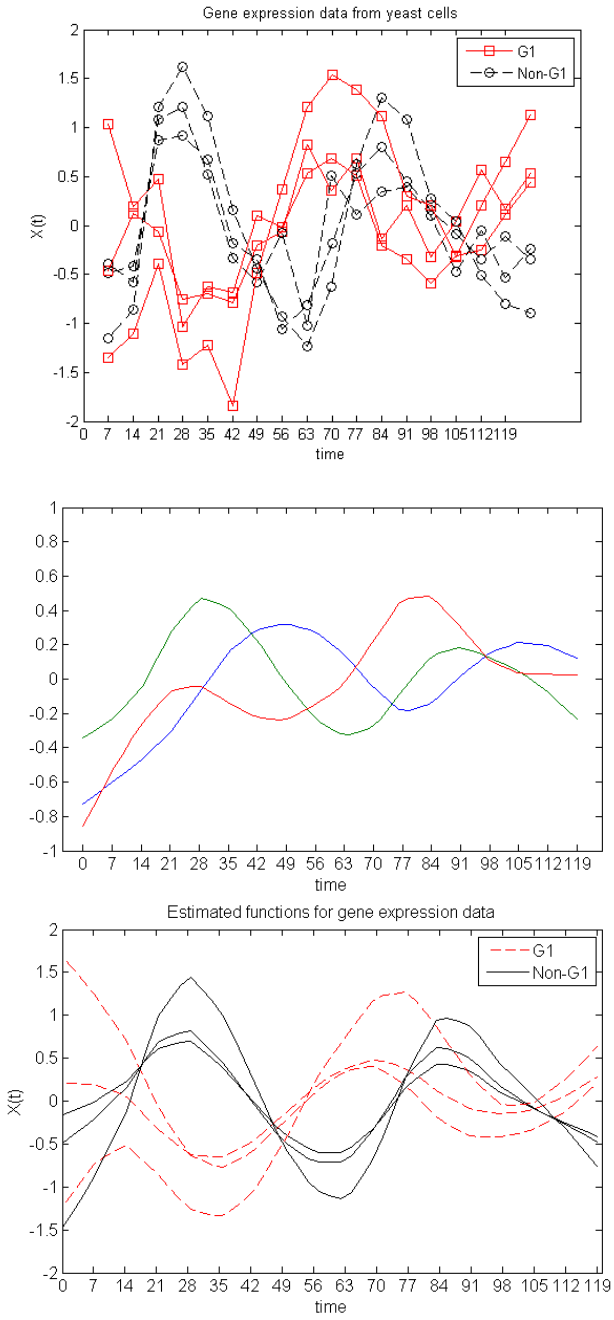


Figure 3.5: Gene expression data: Raw data, estimated FPC curves, and smoothed estimates of data

3.6 Discussion

In this section, we propose two methods analogous to linear discriminant analysis for classifying curves. These methods can be applied in the two-class case as well as cases where $K > 2$. They are flexible and can be applied to sparse and irregularly sampled curves to good effect. Through simulation, we have shown that these methods are effective and compare favorably to existing methods in the literature, showing modest gains in classification error rate. These results are consistent with our motivating examples from the literature, where we find our methods competitive with or superior than the alternatives in all three cases.

Determining which of the two methods we propose is preferable for a given problem is not obvious. From the simulation results, the "centered" method is clearly susceptible to the special case where the difference in the mean functions is not in the space of the estimated eigenfunctions. For data of this type, the results are substantially worse for the the "centered" method than all of the alternatives, including the "uncentered" method. When analyzing data, it may therefore be preferable to err on the side of caution and use the "uncentered" method. However, when we look at examples from the literature, none of our three cases show the "centered" method performing dramatically worse than the alternatives. Even for the spinal bone mineral density example where the "uncentered" method is clearly the better of the two, the "centered" method is about on par with FLDA. It could be that for real data, the special situation described in Case 2 simulation may not be realistic.

For both of the methods, we assume that the covariance operator is identical across all groups. This allows us to pool our data when estimating it. We also assume that the PC scores are normally distributed. While both of these assumptions are popular for scalar data (they are analogs of assumptions made for LDA), it is unclear how robust our method is to their violation.

When normality is violated, it has been shown that LDA performs poorly relative to logistic regression for scalar data, and this may also be the case for functional data.

Chapter 4

Functional Nonlinear Regression

4.1 Introduction

In clinical trials, data is often characterized by a single response variable along with one or more predictor variables observed repeatedly at different time points. This is typically referred to as longitudinal data, and there is a substantial literature discussing longitudinal data analysis. See Rigby (2003) for an introduction to the subject. Often times, these repeated longitudinal measurements are made at irregular time intervals, and for some patients, the total number of observations made can be relatively small. Data fitting this description is referred to as sparse and irregularly sampled longitudinal data.

A more modern approach is to treat these repeated observations as specific realizations of some underlying process and then try to model the process. This is the approach used in functional data analysis. Examples of functional data are prevalent in the literature. For example, the Bal-

timore Longitudinal Study of Aging measured systolic blood pressure of patients at each visit to the Gerontology Research center. (Shock et al., 1984) If we think of a patient's blood pressure as a continuous function over the patient's age, we can treat the blood pressure observed at each patient visit as realizations of this smooth underlying process. Since patients entered and left the study at different times and many patients missed appointments, the observations are often sparsely sampled and sampled at irregular time points.

Analyzing functional data requires different techniques than are traditionally used to analyze longitudinal data. A general overview of functional data analysis including some fundamental methods was given by Ramsay and Silverman (2005). While most models are parametric, some nonparametric methods for functional regression have also been discussed in the literature. Kadri et al. (2009) use functional reproducing kernel Hilbert spaces (RKHSs) to solve the nonparametric regression problem

$$y(t) = f(x(t)) + \epsilon(t), \tag{4.1}$$

where both the predictor and response are functions. Ferraty et al. (2007) adapt the Nadaraja-Watson estimator to the functional data setting to produce a nonparametric functional regression estimator. Lian (2007) discuss nonlinear regression in the RKHS setting. However, these methods generally assume that the predictor curves are fully observed. Often times, data is sparse and irregularly sampled as in the BLSA.

Parametric models capable of handling such data are well known. Yao et al. (2005a) introduces a technique for functional linear regression that can be used with sparse and irregular data. They have extended this model to address functional quadratic regression as well as higher-order

polynomials in Yao and Müller (2010). In both cases, their analysis uses PACE (Yao et al., 2005b) to estimate the full curves. While the general order polynomial model is much more flexible than the simple linear model, it still relies on a parametric structure to approximate the relationship between the predictor and response.

We propose a nonlinear method that removes these model assumptions but can still be applied to sparse and irregularly sampled data. Lian (2007) and Kadri et al. (2009) motivate our work, as they discuss similar methods but rely on fully observing the predictor curve. The remainder of this chapter is organized as follows. In Section 2, we discuss existing parametric methods for sparse and irregularly sampled data in more detail. Section 3 introduces our nonlinear method, and Section 4 discusses tuning. Sections 5 and 6 compare our method to parametric models in simulated examples as well as the BLSA data, and we conclude with a brief discussion in Section 7.

4.2 Parametric Methods

Two prominent methods for modeling functional data with a scalar response are Functional Linear Regression (Yao et al., 2005a) and Functional Quadratic Regression (Yao and Müller, 2010).

The functional linear regression model is

$$Y_i = \beta_0 + \int_{\mathcal{T}} \beta_1(t) X_i^C(t) dt + \epsilon_i, \tag{4.2}$$

where $X_i^C(t) = X_i(t) - \mu(t)$ is the ith curve after centering, and ϵ_i is random error with mean 0 and finite variance. Note that $\beta_0 = \mu_y + \int_{\mathcal{T}} \beta(t)\mu(t)dt$.

The more flexible functional quadratic model is given by

$$Y_i = \beta_0 + \int_{\mathcal{T}} \beta_1(t)X_i^C(t)dt + \int_{\mathcal{T}}\int_{\mathcal{T}} X^C(s)\beta_2(s,t)X_i^C(t)dsdt + \epsilon_i. \tag{4.3}$$

The additional functional quadratic term allows us to model more complex relationships between the predictor function and the response. Details are given in Yao and Müller (2010) for extending this model for higher-order terms, which allows for further flexibility. This generalization can be called functional polynomial regression.

Regardless of how many polynomial terms are used, prediction of the coefficient functions relies on an eigenfunction expansion. Using the eigenfunction basis, each predictor curve can be written using the Karhunen-Loève representation as

$$X_i(t) = \mu(t) + \sum_{l=1}^{\infty} \xi_{il}\phi_l(t). \tag{4.4}$$

The FPC scores can be written as

$$\xi_{il} = \int_{\mathcal{T}} X_i^C(t)\phi_l(t)dt \tag{4.5}$$

for each $l = 1, 2, \ldots$. Then $E(\xi_l) = 0$ and $var(\xi_l) = \lambda_l$ for each l, and the scores are uncorrelated with one another. This basis expansion must be truncated after K terms so that estimation

can take place. Yao et al. (2005a) use AIC to choose K, while Yao and Müller (2010) use BIC.

4.3 Functional Nonlinear Regression

The functional nonlinear regression model is

$$Y_i = F(X_i(t)) + \epsilon_i. \tag{4.6}$$

The only assumptions we make are that the errors, ϵ_i are uncorrelated with mean 0 and finite variance, and that $F : L_2(\mathcal{T}) \to \mathbb{R}$. Here, $L_2(\mathcal{T})$ denotes the set of square integrable functions on $\mathcal{T}$.

Our method is designed to address sparse and irregularly sample data that is typical of longitudinal studies such as the BLSA. Therefore, we assume that for each response, Y_i, we observe a curve $X_i(t)$ at N_i random time points, denoted $T_{i1}, ..., T_{iN_i}$, on some domain, $\mathcal{T}$, where N_i is a random variable independent of the other random variables defining the number of observations for curve i. We assume $X_i(t)$ is square integrable on $\mathcal{T}$, and we allow for observations to be made with measurement error, denoted δ_{ij}. Let the observed value for the ith curve at time T_{ij} be $U_{ij} = X_i(T_{ij}) + \delta_{ij}$. The measurement errors are assumed be independent both between and within curves with mean 0 and $var(\delta_{ij}) < \infty$. We define a mean function, $\mu(t) = E(X_i(t))$, and a covariance operator, $V(s,t) = cov(X_i(s), X_i(t))$. Define the eigenfunctions and eigenvalues of $V(s,t)$ as $\phi_1(t), \phi_2(t), \ldots$ and $\lambda_1, \lambda_2, \ldots$ respectively, with $\lambda_1 \geq \lambda_2 \geq \ldots$. Then we can write

$$V(s,t) = \sum_{j=1}^{\infty} \lambda_l \phi_l(s) \phi_l(t). \tag{4.7}$$

We observe

$$U_{ij} = \mu(T_{ij}) + \sum_{l=1}^{\infty} \xi_{il} \phi_l(T_i j) + \delta_{ij}, \;\; j = 1, \ldots, N_i. \tag{4.8}$$

Therefore, the ith observation is $\{Y_i, (T_{ij}, U_{ij}), j = 1, \ldots, N_i; i = 1, \ldots, n\}$.

We use PACE (Yao et al., 2005b) to estimate the true underlying curves, $X_i(t)$. PACE first estimates $\mu(t)$ and $V(s,t)$ using a nonparametric technique known as local linear smoothing. See Fan and Gijbels (1996) for a discussion of these methods. Then, Functional Principal Components is applied to $\widehat{V}(s,t)$ to obtain $\widehat{\phi}_l(t)$ and $\widehat{\lambda}_l$, $l = 1, \ldots, K$, where K has been chosen via BIC. An estimate for the expectation of the FPC scores conditional on the observed time points is found by plugging these estimates into

$$\tilde{\xi}_{il} = E(\xi_{il} | \boldsymbol{U}_i, \boldsymbol{T}_i) = \lambda_l \boldsymbol{\phi}_{il}^T \Sigma_{U_i}^{-1} (\boldsymbol{U}_i - \boldsymbol{\mu}_i). \tag{4.9}$$

With the trajectories estimated, we can fit the functional nonlinear regression model. To estimate $F(t)$, we solve

$$\min_{F \in \mathcal{F}} \sum_{i=1}^{N} (Y_i - F(\widehat{X}_i(T)))^2 + \gamma ||F||, \tag{4.10}$$

where $\gamma \geq 0$ penalizes the roughness of the function, F, and $\mathcal{F}$ is some function space. We

define $\mathcal{F}$ to be a reproducing kernel Hilbert space (RKHS).

Let $K(\cdot,\cdot)$ be a bivariate kernel function from $L_2(\mathcal{T}) \times L_2(\mathcal{T})$ to $\mathbb{R}$. One such kernel is the Gaussian kernel, defined as

$$K(\widehat{X}_i(\cdot), \widehat{X}_j(\cdot)) = exp\left\{-||\widehat{X}_i(\cdot) - \widehat{X}_j(\cdot)||_2^2/\rho^2\right\} \tag{4.11}$$

where

$$||\widehat{X}_i(\cdot) - \widehat{X}_j(\cdot)||_2 = \sqrt{\int_{\mathcal{T}} (\widehat{X}_i(t) - \widehat{X}_j(t))^2 dt}. \tag{4.12}$$

Then we define $\mathcal{F}_K$ as the RKHS defined by $K(\cdot,\cdot)$. Then with $\mathcal{F} = \mathcal{F}_K$, the solution to Eq. 4.10 is of the form $F(X(t)) = c_0 + \sum_{j=1}^n c_j K(X(\cdot), \widehat{X}_j cdot))$. This is a result of the representer theorem. (See Wahba (1990) and Theorem 1 from Lian (2007) for a generalization.) Our problem has been reduced to finding $\widehat{c}_0, \widehat{c}_1, \ldots, \widehat{c}_n$ by solving

$$\min_{c_0,c_1,\ldots,c_n} \sum_{i=1}^n \left[Y_i - c_0 - \sum_{j=1}^n c_j K(\widehat{X}_i(\cdot), \widehat{X}_j(\cdot)) \right]^2 + \gamma \sum_{i=1}^n \sum_{j=1}^n c_i K(\widehat{X}_i(\cdot), \widehat{X}_j(\cdot)) c_j. \tag{4.13}$$

Thus, our estimate for $F(X(t))$ is

$$\widehat{F}(X(t)) = \widehat{c}_0 + \sum_{i=1}^n \widehat{c}_i K(\widehat{X}_i(\cdot), X(\cdot)). \tag{4.14}$$

.

This RKHS-based solution is very general, and both functional linear regression and functional quadratic regression are special cases. For example, by taking the linear kernel, $K(\widehat{X}_i(\cdot), \widehat{X}_j(\cdot)) = \int_{t\in\mathcal{T}} \widehat{X}_i(t)\widehat{X}(t))dt$, Eq. 4.14 reduces to functional linear regression when the regularization parameter γ is set to 0.

Prediction for new curves is straightforward. Since we assume sparse and irregular observations, for the new curve, $X^*(t)$ we observe values $U_j^* = X^*(T_j) + \epsilon_j^*$, $j = 1, \ldots, N^*$. We use PACE to estimate ξ_l^* for $l = 1, \ldots, K$ for the estimated eigenfunctions, $\widehat{\phi}_l(t)$ using $\widehat{\lambda}_l$ and $\widehat{\mu}(t)$. Denote this estimate of $X^*(t)$ as $\widehat{X}^*(t)$. Then our predicted response is

$$\widehat{Y}_i^* = \widehat{F}(\left\{(U_j^*, T_j^*), j = 1, \ldots, N^*\right\}) = \widehat{c}_0 + \sum_{i=1}^{n} c_i K(\widehat{X}_i(\cdot), \widehat{X}^*(\cdot)). \tag{4.15}$$

4.4 Tuning

Two tuning steps are necessary for this RKHS-based functional nonlinear regression procedure. First, we must choose the number of basis functions to use when estimating the predictor trajectories using functional principal components. Yao et al. (2005b) recommend using BIC for choosing the optimal number of functional principal components, so this is what we use in the examples below. Alternatives like thresholding and AIC are also possible, but we do not explore them here.

Secondly, we need to tune the regularization parameter, γ, used to smooth the final function. We tune this using K-fold cross-validation over squared prediction error. The squared prediction error for the ith observation is

$$(Y_i - \widehat{F}(\{(U_{ij}), T_{ij}, j = 1, \ldots, N^*\})^2. \tag{4.16}$$

Let $S_1, S_2, \ldots, S_K$ be randomly selected sets such that $S_1 \cup S_2 \cup \ldots \cup S_K = \{1, 2, \ldots, n\}$ and $S_k \cap S_m = \emptyset$, where $1 \leq m \neq k \leq n$. Let $S_k^C = \{1, 2, \ldots, n\}/S_k$ be the compliment of S_k. Then for a given γ, the K-fold cross-validation squared prediction error is

$$Error_\gamma = \sum_{k=1}^{K} \sum_{i \in S_k} (Y_i - \widehat{F}^{\gamma}_{S_k^C}(\{(U_{ij}, T_{ij}), j = 1, \ldots, N^*\})^2, \tag{4.17}$$

where $\widehat{F}^{\gamma}_{S_k^C}$ denotes the estimated functional nonlinear regression function for the training data set, S_k^C, and the smoothing parameter, γ. We choose a grid of candidate values for γ and choose the candidate that minimizes Eq. 4.17.

4.5 Simulation

We generate data using three eigenfunction,

$$\phi_1(t) = -\sqrt{\frac{1}{5}} cos\left(\frac{\pi t}{5}\right) \tag{4.18}$$

$$\phi_2(t) = \sqrt{\frac{1}{5}} sin\left(\frac{\pi t}{5}\right) \tag{4.19}$$

$$\phi_3(t) = -\sqrt{\frac{1}{5}}\left(\frac{\pi 2t}{5}\right), \tag{4.20}$$

defined over the domain $\mathcal{T} = [0, 10]$. Each curve, $X_i(t) = \mu_x(t) + \sum_{l=1}^{3} \xi_{il}\phi_l(t)$, where $\xi_{il} \sim$

$N(0, \lambda_l)$ and $\mu_x(t) = t + sin(t)$. We choose $\boldsymbol{\lambda} = [4\ 2\ 1]$. For $i = 1, \ldots, n$, we observe $U_i(T_{ij}) = X_i(T_{ij}) + \epsilon_{ij}$, where $j = 1, \ldots, N_i$ and $\epsilon \sim N(0, .5^2)$.

Two models are considered:

$$\begin{aligned} F_1(X(t)) =& 1 + .5\left|\left|\int_{\mathcal{T}}(X(t) - \mu(t))\phi_1(t)dt\right| - 2\right| + .5\left|\int_{\mathcal{T}}(X(t) - \mu(t))\phi_2(t)dt\right| + \\ & .5sin\left(2\pi\int_{\mathcal{T}}(X(t) - \mu(t))\phi_3(t)dt\right) \end{aligned} \tag{4.21}$$

and

$$\begin{aligned} F_2(X(t)) =& \frac{1}{3}\sum_{m=1}^{3}\int_{\mathcal{T}}(X(t) - \mu(t))\phi_m(t)dt + \frac{1}{3}\sum_{m=1}^{3}\left(\int_{\mathcal{T}}(X(t) - \mu(t))\phi_m(t)dt\right)^2 + \\ & \frac{1}{3}\int_{\mathcal{T}}(X(t) - \mu(t))\phi_1(t)dt\int_{\mathcal{T}}(X(t) - \mu(t))\phi_2(t)dt. \end{aligned} \tag{4.22}$$

Case 1 is a nonlinear model while Case 2 fits the quadratic regression model proposed in Yao and Müller (2010). In terms of the FPC scores, $F_1(X(t)) = 1 + .5||\xi_1| - 2| + .5|\xi_2| + .5\sin(2\pi\xi_3)$ and $F_2(X(t)) = \frac{1}{3}(\xi_1 + \xi_2 + \xi_3 + \xi_1^2 + \xi_2^2 + \xi_3^2 + \xi_1\xi_2)$.

For each of these models, a sparse case and a regularly sampled case are considered. For the latter, $N_i = N = 31$ for all i, and $T_j = \frac{j-1}{3}$ for $j = 1, \ldots, 31$. For the sparse case, N_i has the discrete uniform distribution on the integer values $4, 5, \ldots, 10$. Given N_i, the T_{ij} are uniformly distributed on $\mathcal{T} = [0, 10]$.

We compare our method to Functional Linear Regression (Yao et al., 2005a) and Functional

Quadratic Regression (Yao and Müller, 2010). Additionally, we compare our method with a simple nonlinear regression method when the data is regularly sampled. For this model, we fit

$$\min_{c_0,c_1,\ldots,c_n} \sum_{i=1}^{n} \left(Y_i - b_0 - \sum_{j=1}^{n} b_j k(\boldsymbol{U}_i, \boldsymbol{U}_j) \right)^2 + \gamma \sum_{i=1}^{n} \sum_{j=1}^{n} b_i k(\boldsymbol{U}_i, \boldsymbol{U}_j) b_j, \tag{4.23}$$

where γ is a regularization parameter and $\boldsymbol{U}_i$ is the vector of realized values from the ith curve. The kernel, $k(\cdot, \cdot)$, is a bivariate kernel function from $\mathbb{R}^{31} \times \mathbb{R}^{31}$ to $\mathbb{R}$. The estimated nonlinear regression function is then

$$\widehat{Y}^* = \widehat{b}_0 + \sum_{j=1}^{n} \widehat{b}_j k(\boldsymbol{U}^*, \boldsymbol{U}_j) \tag{4.24}$$

For both nonlinear regression (NR) and our proposed functional nonlinear regression (FNR) method, we use the Gaussian kernel. In both cases, the tuning parameter, γ is chosen using 5-fold cross-validation over the grid of points, $\{.25, .5, .75, 1, 1.25, 1.5\}\tau$, where τ regularizes over the median 2-norm distance between the observed curves. For FNR, τ is the median of $\{||\widehat{X}_i(\cdot) - \widehat{X}_j(\cdot)||_2,\ 1 \leq i, j \leq n\}$, and for NR, τ is the median of $\{||\boldsymbol{U}_i, \boldsymbol{U}_j||_2\ 1 \leq i, j \leq n\}$.

For each simulation, $n = 200$ training data points are generated. An independent data set of $n_{test} = 1000$ is used as test data. Average squared prediction error over $n_{sim} = 400$ is reported in Figure 4.1 for both regular and sparse data sets and for both models, $F_1(X(t))$ and $F_2(X(t))$. Note that the NR method is only used for the regular data case. For both cases, when the true relationship is nonlinear (Model 1), FNR shows best performance, and in the "regular data" case the simple nonlinear method actually outperforms the parametric functional models. For the two cases where $F_2(\cdot)$ is used, the FQR shows the best performance, although FNR is still

competitive. This is reasonable to expect, since for this case, the quadratic model is the oracle.

4.6 Real Data

We also apply our method to real data from the Baltimore Longitudinal Study of Aging (BLSA). Our data set consists of 1590 male individuals of ages ranging from the 20s to the 90s. Visits were scheduled biannually, but individuals frequently missed visits. This means that some individuals are not sampled frequently (sparse sampling) and that all individuals are not sampled at the same time points (irregular sampling). This type of data is typical for longitudinal studies over long time frames that rely on individuals making attending their appointments.

While many variables were sampled in this study, we focus on systolic blood pressure. We use information on blood pressure from ages 40 through 45 to predict blood pressure from ages 46 through 50. Data is available for 250 of these individuals over this age range, and their trajectories are plotted in Figure 4.2a. We split the data randomly into a training set of 200 subjects and a test set of 50 subjects whose data is not used to fit the model. Estimates for the first two principal components from PACE are plotted in Figure 4.2b.

Since the data are sparse and irregular, we cannot apply the nonlinear regression method discussed above. Therefore, we apply functional linear regression (FLR), functional quadratic regression (FQR), and our functional nonlinear regression method (FNR). Applying the fit models to the test data set, we get average squared prediction errors of 138.6713 (FLR), 138.1681 (FQR), and 131.1970 (FNR). We see that the more flexible FQR outperformed FLR by 0.5032 and that the even-more-flexible nonparametric FNR method outperformed FQR by 7.4743.

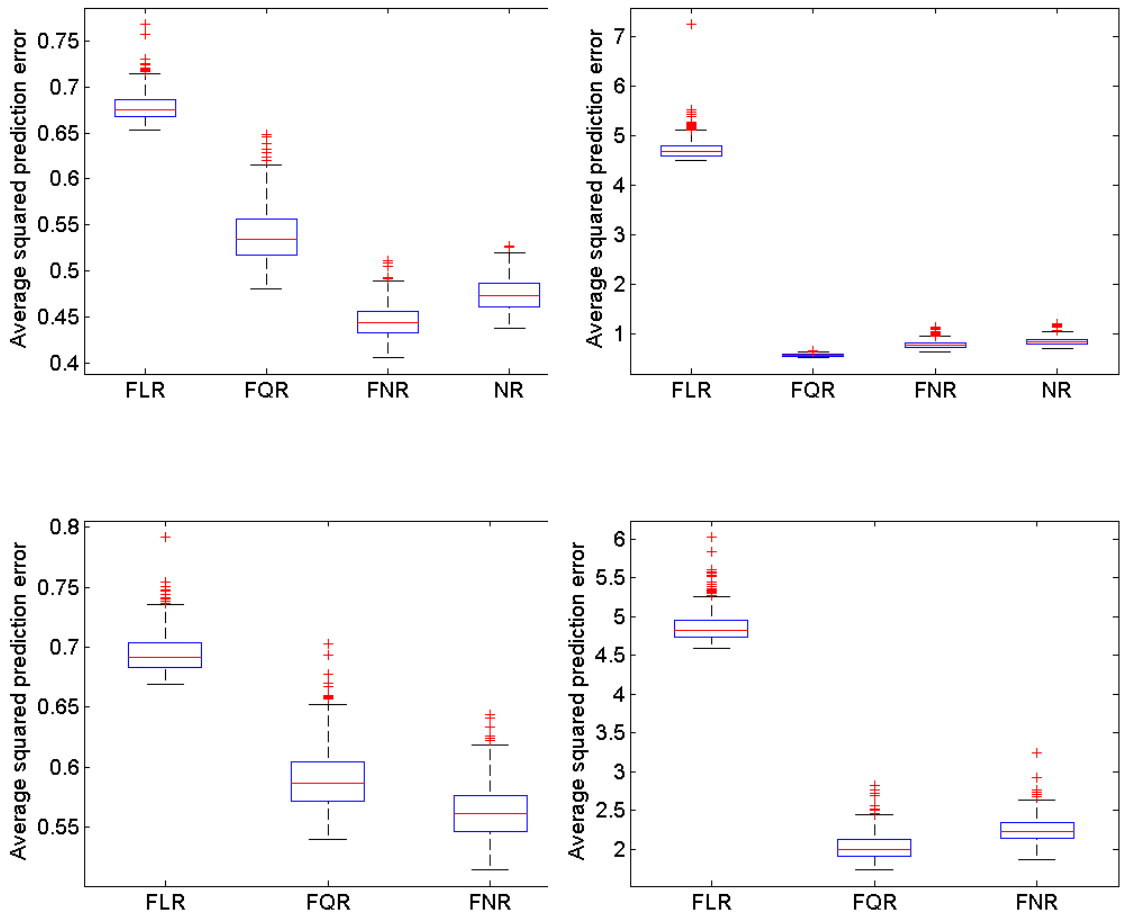
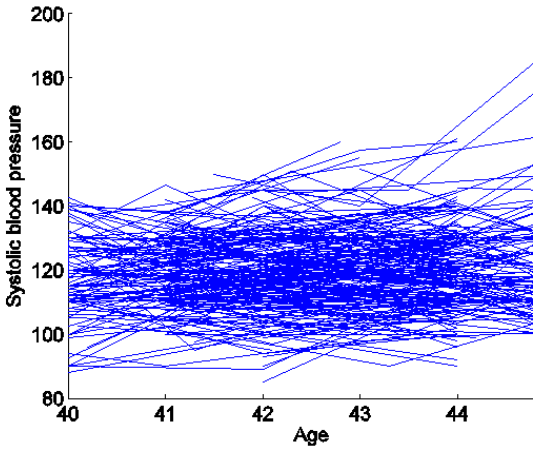
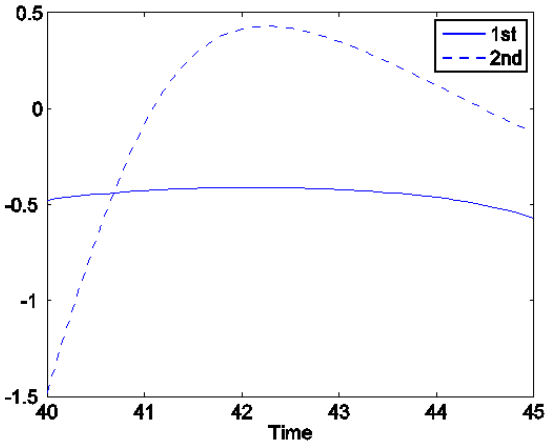


Figure 4.1: Boxplots of average squared prediction errors. The top row is the regular case, and the bottom row is the sparse case. The first column uses model F_1 and the second column uses model F_2.



(a) Observed BP for subjects ages 40 to 45 from BLSA



(b) First two eigen functions

Figure 4.2: BLSA Data

4.7 Discussion

The RKHS-based nonlinear regression method proposed above is a flexible method for modeling complex relationships between functional predictor variables and scalar responses. Unlike other parametric methods such as functional linear regression and functional quadratic regression, no parametric model is specified. Unlike previous nonparametric methods, functional nonlinear regression does not rely on fully observed predictor curves. By using PACE to estimate the full curve, this method can be applied to sparse and irregular data like the Baltimore Longitudinal Study of Aging. In simulation and on real data, functional nonlinear regression outperformed parametric methods in terms of average squared prediction error.

REFERENCES

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716 – 723, Dec 1974.

Laura K. Bachrach, Trevor Hastie, May-Choo Wang, Balasubramanian Narasimhan, and Robert Marcus. Bone mineral acquisition in healthy asian, hispanic, black, and caucasian youth: A longitudinal study. *Journal of Clinical Endocrinology & Metabolism*, 84(12):4702–4712, 1999. doi: 10.1210/jc.84.12.4702. URL `http://jcem.endojournals.org/content/84/12/4702.abstract`.

P. E. Castro, W. H. Lawton, and E. A. Sylvestre. Principal modes of variation for processes with continuous sample curves. *Technometrics*, 28(4):pp. 329–337, 1986. ISSN 00401706. URL `http://www.jstor.org/stable/1268982`.

Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008. doi: 10.1093/biomet/asn034.

William Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

Jianqing Fan and Irne Gijbels. Variable bandwidth and local linear regression smoothers. *Annals of Statistics*, 20(4):2008–2036, 1992.

Jianqing Fan and Irne Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall, 1996.

Frédéric Ferraty, André Mas, and Philippe Vieu. Nonparametric regression on functional data: Inference and pracitcal aspects. *Australian & New Zealand Journal of Statistics*, 49(3):267–286, 2007.

P. J. Green and B. W. Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. Chapman and Hall, London, 1994.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and Ebooks Corporation. *The Elements of Statistical Learning*. Springer, Dordrecht, 2009.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Jian Huang, Shuangge Ma, and Cun hui Zhang. Adaptive lasso for sparse highdimensional regression. Technical report, University of Iowa, 2006.

Gareth M. James and Trevor J. Hastie. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(3):pp. 533–550, 2001. ISSN 13697412. URL http://www.jstor.org/stable/2680587.

Gareth M. James, Jing Wang, and Ji Zhu. Functional linear regression that's interpretable. *The Annals of Statistics*, 37:2083, 2009.

GM James, TJ Hastie, and CA Sugar. Principal component models for sparse functional data. *Biometrika*, 87(3):587–602, 2000. doi: 10.1093/biomet/87.3.587.

Hachem Kadri, Emmanuel Duflos, Manuel Davy, Philippe Preux, and Stephane Canu. General Framework for Nonlinear Functional Regression with Reproducing Kernel Hilbert Spaces. Research Report RR-6908, INRIA, 2009. URL http://hal.inria.fr/inria-00378381.

Eun Ryung Lee and Byeong U. Park. Sparse estimation in functional linear regression. *Journal of Multivariate Analysis*, 105(1):1 – 17, 2012. ISSN 0047-259X. doi: 10.1016/j.jmva.2011.08.005.

Xiaoyan Leng and Hans-Georg Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76, 2006. doi: 10.1093/bioinformatics/bti742.

Bin Li and Qingzhao Yu. Classification of functional data: A segmentation approach. *Computational Statistics &; Data Analysis*, 52(10):4790 – 4800, 2008. ISSN 0167-9473. doi: 10.1016/j.csda.2008.03.024.

Heng Lian. Nonlinear functional models for functional responses in reproducing kernel hilbert spaces. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 35(4):pp. 597–606, 2007. ISSN 03195724. URL http://www.jstor.org/stable/20445281.

Alberto Muñoz and Javier González. Representing functional data using support vector machines. *Pattern Recognition Letters*, 31(6):511 – 516, 2010. ISSN 0167-8655. doi: 10.1016/j.patrec.2009.07.014.

S original by Berwin A. Turlach R port by Andreas Weingessel. *quadprog: Functions to solve Quadratic Programming Problems.*, 2011. URL http://CRAN.R-project.org/package=quadprog. R package version 1.5-4.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):pp. 539–572, 1991. ISSN 00359246. URL http://www.jstor.org/stable/2345586.

James O Ramsay and B W Silverman. *Functional Data Analysis (2nd Edition)*. Springer, 2005.

Opgen R. Rhein and K. Strimmer. Inferring gene dependency networks from genomic longitudinal data: a functional data approach. *REVSTAT*, 4:53–65, 2006.

John Rice. Bandwidth choice for nonparametric regression. *The Annals of Statistics*, 12(4):pp. 1215–1230, 1984. ISSN 00905364. URL http://www.jstor.org/stable/2240998.

John A. Rice and Colin O. Wu. Nonparametric mixed effects models for unequally sampled noisy curves. *Biometrics*, 57(1):253–259, 2001. ISSN 1541-0420.

Alan S. Rigby. Review: Analysis of Longitudinal Data by P. J. Diggle; P. J. Heagerty; S. L. Zeger; K. -Y. Liang. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 52 (2):239–240, 2003.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

Nathan W. Shock, Richard C. Greulich, Jr. Costa Paul T, Reubin Andres, Edward G. Lakatta, David Arenberg, and Jordan D. Tobin. Normal human aging: The baltimore longitudinal study on aging, 1984. URL http://health-equity.pitt.edu/2557/.

Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycleregulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00490.x.

G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990. URL http://www.ams.org/mathscinet-getitem?mr=1045442.

Gordon Smyth with contributions from Yifang Hu, Peter Dunn, and Belinda Phipson. *statmod: Statistical Modeling*, 2011. URL http://CRAN.R-project.org/package=statmod. R package version 1.4.14.

Fang Yao and Hans-Georg Müller. Functional quadratic regression. *Biometrika*, 97(1):49–64, 2010. doi: 10.1093/biomet/asp069. URL http://biomet.oxfordjournals.org/content/97/1/49.abstract.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33:2873–2903, 2005a.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005b. doi: 10.1198/016214504000001745. URL http://pubs.amstat.org/doi/abs/10.1198/016214504000001745.

Daowen Zhang, Xihong Lin, and Maryfran Sowers. Two-Stage Functional Mixed Models for Evaluating the Effect of Longitudinal Covariate Profiles on a Scalar Outcome. *Biometrics*, 63(2):351–362, 2007.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

APPENDICES

Appendix A

Fused LASSO Model

The fused LASSO model is

$$L_{\lambda_1,\lambda_2}(y, X, \beta) = .5(y - X\beta)^T(y - X\beta) + \lambda_1 \sum_{k=1}^{p} |\beta_k| + \lambda_2 \sum_{k=2}^{p} |\beta_k - \beta_{k-1}|, \qquad \text{(A.1)}$$

where β is a $p \times 1$ vector of predictors, X is a $n \times p$ standardized design matrix, and y is a $n \times 1$ response vector. This model can be fit using Quadratic Programming, which solves problems of the form

$$\arg\min(.5d^T Db - d^T b) \quad s.t.\ A^T b \geq b_0. \qquad \text{(A.2)}$$

We can re-write the Fused LASSO model

$$D = \begin{pmatrix} X^T X & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix};\ d = \left(Y^T X, -\lambda_1 \mathbf{1}, -\lambda_2 \mathbf{1}\right);\ A^T = \begin{pmatrix} -I & I & 0 \\ I & I & 0 \\ Q & 0 & I \\ -Q & 0 & I \end{pmatrix};\ b = (0), \tag{A.3}$$

where

$$Q = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{pmatrix}. \tag{A.4}$$

D is $3p \times 3p$, d is $3p \times 1$, A^T is $(4p-2) \times (3p-1)$, b is $(4p-2) \times 1$, and Q is $p \times (p-1)$. For our purposes here, p is the number of basis functions used and may not necessarily be the number of intervals into which the domain of $X(t)$, T, is broken into. For example, the fit using the basis functions described in Equation 2.4 requires an "intercept" basis function, $\phi_0(t) = \int_0^1 X(t)dt$, so when calculating the dimensions of the matrices for the Fused LASSO fit of data constructed using this data, we should substitute $(p+1)$ for p. This above quadratic programming solution can be calculated for any given non-negative values of λ_1 and λ_2.

Appendix B

Additional Simulation Results

Error Type	Fit	n=50	100	200	1000
MISE 1	Piecewise Constant	0.21	0.18	0.17	0.16
	Ramp	0.15	0.10	0.09	0.07
	Fourier	0.59	0.32	0.18	0.16
	FLiRTI ($d = 2$)	0.13	0.10	0.08	0.05
MISE 2	Piecewise Constant	0.09	0.05	0.02	0.01
	Ramp	0.41	0.04	0.02	0.01
	Fourier	0.11	0.06	0.03	0.01
	FLiRTI ($d = 2$)	0.12	0.06	0.03	0.01
Type I error	Piecewise Constant	0.31	0.35	0.43	0.48
	Ramp	0.19	0.12	0.16	0.13
	Fourier	0.28	0.28	0.31	0.29
	FLiRTI ($d = 2$)	0.25	0.21	0.17	0.14
Type II error	Piecewise Constant	0.17	0.16	0.14	0.11
	Ramp	0.17	0.16	0.15	0.15
	Fourier	0.23	0.24	0.19	0.18
	FLiRTI ($d = 2$)	0.11	0.07	0.05	0.03

Table B.1: Simulation results for triangle coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	1000
MISE 1	Piecewise Constant	0.19	0.18	0.17	0.17
	Ramp	0.17	0.15	0.12	0.11
	Fourier	0.50	0.25	0.15	0.08
	FLiRTI ($d = 2$)	0.34	0.26	0.32	0.22
MISE 2	Piecewise Constant	0.11	0.05	0.02	0.01
	Ramp	0.42	0.07	0.03	0.01
	Fourier	0.11	0.06	0.03	0.01
	FLiRTI ($d = 2$)	0.12	0.06	0.03	0.01
Type I error	Piecewise Constant	0.15	0.27	0.42	0.70
	Ramp	0.09	0.23	0.27	0.44
	Fourier	0.29	0.31	0.28	0.30
	FLiRTI ($d = 2$)	0.07	0.05	0.03	0.03
Type II error	Piecewise Constant	0.19	0.17	0.12	0.06
	Ramp	0.20	0.15	0.13	0.08
	Fourier	0.22	0.22	0.20	0.16
	FLiRTI ($d = 2$)	0.15	0.12	0.12	0.10

Table B.2: Simulation results for Valley coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	1000
MISE 1	Piecewise Constant	1.01	0.94	0.81	0.30
	Ramp	0.64	0.64	0.47	0.21
	Fourier	1.08	0.72	0.57	0.16
	FLiRTI ($d = 2$)	1.12	1.01	0.88	0.21
MISE 2	Piecewise Constant	0.10	0.05	0.02	0.01
	Ramp	0.08	0.05	0.02	0.28
	Fourier	0.09	0.05	0.02	0.01
	FLiRTI ($d = 2$)	0.10	0.05	0.03	0.01
Type I error	Piecewise Constant	0.26	0.19	0.19	0.07
	Ramp	0.27	0.30	0.21	0.19
	Fourier	0.35	0.29	0.30	0.14
	FLiRTI ($d = 2$)	0.22	0.18	0.19	0.12
Type II error	Piecewise Constant	0.00	0.00	0.00	0.19
	Ramp	0.00	0.01	0.01	0.24
	Fourier	0.04	0.02	0.01	0.21
	FLiRTI ($d = 2$)	0.07	0.06	0.04	0.17

Table B.3: Simulation results for Step coefficient function for different values of n with EBIC tuning. ($p = 35$)