12-2017

Functional data analysis methods for predicting disease status.

Sarah Kendrick
University of Louisville

FUNCTIONAL DATA ANALYSIS METHODS FOR PREDICTING DISEASE STATUS

By

Sarah Kendrick
B.A., Wittenberg University, 2010
M.S., University of Louisville, 2013

A Dissertation
Submitted to the Faculty of the
School of Public Health and Information Sciences
of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Biostatistics

Department of Bioinformatics and Biostatistics
University of Louisville
Louisville, Kentucky

December, 2017

FUNCTIONAL DATA ANALYSIS METHODS FOR PREDICTING DISEASE STATUS

By

Sarah Kendrick
B.A., Wittenberg University, 2010
M.S., University of Louisville, 2013

A Dissertation Approved on

November 20, 2017

by the following Dissertation Committee:

Dr. Qi Zheng, Dissertation Director

Dr. Guy Brock, Dissertation Co-director

Dr. K.B. Kulasekera

Dr. Jeremy Gaskins

Dr. Maiying Kong

Dr. Nichola Garbett

DEDICATION

This dissertation is dedicated to my parents who have given me invaluable educational opportunites, been my biggest supporters, and greatest motivators.

ACKNOWLEDGMENTS

A special thank you to my dissertation committe members: Dr. Qi Zheng, Dr. Guy Brock, Dr. KB Kulasekera, Dr. Nichola Garbett, Dr. Jeremy Gaskins, and Dr. Maiying Kong for all of your time, dedication, and support throughout this process.

Also, thank you, to Dr. Ruoqing Zhu and his student Miao Ruizhong for their collaboration on the Local Basis Random Forests for Functional Data project.

ABSTRACT

FUNCTIONAL DATA ANALYSIS METHODS FOR PREDICTING DISEASE STATUS

Sarah Kendrick

November 20, 2017

Introduction: Differential scanning calorimetry (DSC) is used to determine thermally-induced conformational changes of biomolecules within a blood plasma sample. Recent research has indicated that DSC curves (or thermograms) may have different characteristics based on disease status and, thus, may be useful as a monitoring and diagnostic tool for some diseases. Since thermograms are curves measured over a range of temperature values, they are often considered as functional data. In this dissertation we propose and apply functional data analysis (FDA) techniques to analyze DSC data from the Lupus Family Registry and Repository (LFRR). The aim is to develop FDA methods to create models for classifying lupus vs. control patients on the basis of the thermogram curves.

Methods: In project 1 we examine how well standard functional regression is able to capture the differences in curves for cases and controls and compare this to a multivariate approach. In project 2 we develop a semiparametric model; the Generalized Functional Partially Linear Single-Index Model (GFPL). This model is useful when there exists some curvature or non-linearity in the logit, which cannot be modeled by the standard Functional Generalized Linear Model (FGLM). It also mitigates the curse of dimensionality, is a more flexible model, and yields interpretable

results. In project 3, we propose a tree-based method: Local Basis Random Forests (LBRF) for Functional Data. This non-parametric method allows us to focus on significant parts of the functional covariates and reduce the noise level.

Results: The standard functional logistic regression model with FPCA scores as the predictors gives an 81.25% correct classification rate on the test data, comparable to results from the multivariate approach. The proposed GFPL gives prediction accuracies and standard errors that are better than the standard FGLM when there is nonlinearity present. The LBRF for functional data yields high prediction accuracy (as high as 97% in simulations and 92% in the Lupus data), especially when the true signal is localized, and is able to capture where the true signal lies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Differential scanning calorimetry (DSC) is used to determine thermally-induced conformational changes of biomolecules within a blood plasma sample. The sample is treated at increasing temperature increments and excess specific heat capacity given off is measured at each temperature. The excess specific heat capacity can be plotted against temperature producing a curve referred to as a thermogram. Recent research has indicated that these curves may have different characteristics based on disease status and, thus, may be useful as a monitoring and diagnostic tool for some diseases[1, 2].

One example where DSC thermograms may be helpful in diagnosis and disease monitoring is with lupus patients. Systemic lupus erythematosus, Lupus, is an auto-immune disease in which individuals' immune systems produce antibodies to cells within the body leading to inflammation. Lupus can affect a wide array of organs/systems within the body and often has symptoms that mimic other diseases. This makes it very difficult to diagnose and monitor Lupus. The American College of Rheumatology provides a list of 11 criteria for potential Lupus diagnosis. An individual is classified as being positive for Lupus if they meet at least 4 of the 11 criteria. This methodology often leads to over-diagnosis, under-diagnosis, and often misses early and mild cases. Therefore, researchers and doctors are looking for new and improved Lupus diagnostic tools [3-5]. We apply our methods to data obtained from the Lupus Family Registry and Repository (LFRR), which consists of 600 de-

identified samples [6]. Plasma samples for 300 patients classified as having Lupus using the ACR criteria were obtained. Another 300 plasma samples from controls without lupus who were matched with diseased individuals based on sex, race, and age were also obtained. However, eight of these were flagged as being poor scans thus we end up with 592 observations; 298 cases and 294 controls.

The idea of using thermograms as a diagnostic tool is still relatively new. Thermograms can be analyzed using multivariate or functional data techniques; most prior work in this area has taken the multivariate approach to analyzing these curves. This dissertation dives into functional analysis of the data with the goal of developing a model(s) that is interpretable and yields high prediction accuracy. We first implement standard functional regression and compare it to multivariate approaches. Then we develop a semiparametric model that extends standard functional regression to allow for more flexibility. We compare this model to the standard functional generalized regression. Finally, we develop a functional extension of the random forest method. This nonparametric method method allows us to focus on significant parts of the functional covariates and reduce the noise level. We compare this model to several other tree-based methods.

CHAPTER 2

APPLICATION AND INTERPRETATION OF FUNCTIONAL DATA ANALYSIS TECHNIQUES TO DIFFERENTIAL SCANNING CALORIMETRY DATA FROM LUPUS PATIENTS

2.1 Introduction

Project one implements several existing Functional Data Analysis (FDA) tools to analyze the DSC data. Functional data analysis is the analysis of curves or functions as a whole. In FDA, the entire curve or function is considered as one unit of observation instead of multiple observations along a time continuum. Ramsay and Silverman described the FDA framework in 2005. Their work shows how to specify basis systems for building up functions, how to build functional data objects, how to smooth functional curves, how to perform functional principal component analysis, and how to implement linear regression within the functional framework [7]. Febrero-Bande and de la Fuente developed a model for functional generalized linear models (FGLM) in 2012 that is capable of handling categorical response variables with functional predictors [8].

The use of thermogram profiles as a diagnostic tool is a relatively new research idea. Few papers have been published in this area and functional data analysis has never been used to analyze thermograms. Fish et al. focused strictly on classification of individuals as cases or controls. They used a non-parametric method and calculated a similarity metric for classification [9]. Fish et al. did not apply functional data

techniques, nor did they explore the effects of covariates on classification. Vega et al. presented a novel method for analyzing thermograms in which they first broke down the thermograms into six individual peaks to represent the curves. They then used parameters corresponding to each peak in a multiparametric comparative method to develop classification criteria [10]. Similar to Fish's paper, Vega et al. only focused on classification, did not implement standard functional data analysis techniques, and did not explore the effect of other covariates on their methods. Finally, Garbett et al. used ANOVA on the first principal component for thermograms along with other covariates to assess the relationship between thermograms, additional covariates, and disease status. They used modified linear discriminant analysis for classification [1]. Garbett et al. used multivariate analysis instead of functional data analysis, and their results are difficult to interpret.

2.2 Methods

The main aim of project 1 is to apply standard Functional Data Analysis techniques to the Lupus DSC data. The techniques we use are regression with a functional response variable and scalar/categorical predictor variable, regression with a scalar/categorical response variable and a functional predictor, and regression with a scalar/categorical response and functional principal component analysis (FPCA) scores as the predictor. Below we describe each method in detail and then illustrate how these methods can be applied to the Lupus data.

2.2.1 Linear Models for Functional Responses

Linear models with a functional response variable and scalar/categorical covariates are used when a researcher is interested in predicting a functional response based on

the values of the covariate(s). The traditional linear model,

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i \tag{2.1}$$

can be extended to the case where the response is now functional. To accomplish this, we first define basis functions. Basis functions and basis expansions are the building blocks of all FDA techniques. The following is an example of a basis function expansion: $x(t) = \sum_{k=1}^{K} c_k\phi_k(t) = \boldsymbol{c}'\boldsymbol{\phi}(t)$. Often, we are interested in a sample of N basis functions; we now have: $x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t)$ for $i = 1, \cdots, N$. Using basis expansions, the model becomes,

$$y_i(t) = \beta_0(t) + \sum_{j=1}^{p} x_{ij}\beta_j(t) + \epsilon_i(t), \tag{2.2}$$

$$\beta_0(t) = \sum_{k=1}^{K} b_{0,k}\phi_k(t), \tag{2.3}$$

$$\beta_1(t) = \sum_{k=1}^{K} b_{1,k}\phi_k(t), ..., \beta_p(t) = \sum_{k=1}^{K} b_{p,k}\phi_k(t), \tag{2.4}$$

Plugging (2.3) and (2.4) into (2.2),

$$y_i(t) = \sum_{k=1}^{K} b_{0,k}\phi_k(t) + \sum_{j=1}^{p}\sum_{k=1}^{K} x_{ij}b_{j,k}\phi_k(t) + \epsilon_i(t), \tag{2.5}$$

where i = 1...n indicates the individual observations, t is the time indicator, K is the number of basis functions used for setting up the basis expansion for $\beta(\cdot)$, ϕ is our matrix of K basis functions, $b_{j,k}$ are the unknown β coefficient vectors; and x_{ij} represents the ijth entry of the design matrix. In the case of the thermogram data, the design matrix will be (m=451 x n=592), with each column containing the excess specific heat capacity values for each individual (n) and each row representing one temperature value (m). This model can also handle additional covariates [11].

Just as in non-functional regression, in functional regression we may be interested in answering some common statistical questions such as,

1. Are the thermograms for cases and controls statistically distinguishable?

2. Are there statistically significant relationships between thermogram profiles and disease status, gender, race, and other covariates?

Functional equivalents of the standard t- and F-tests can be performed to answer such questions. Due to the fact that functional data are inherently high-dimensional, permutation tests are used to determine the critical values for these tests (See Ramsay & Silverman 2009 for details) [11].

2.2.2 Generalized Linear Models with Functional Covariates

Generalized linear models are often used in the presence of a categorical response. Here, as we are interested in using DSC profile to predict disease status, we consider functional logistic regression. In this case, the link function is the logit function,

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \int x_i(t)\alpha_1(t)dt, \tag{2.6}$$

where

$$x_i(t) \approx \sum_{l=1}^{L} \phi_l(t)c_{i,l} = \boldsymbol{\phi(t)'c}_i, \tag{2.7}$$

$$\alpha_1(t) \approx \sum_{k=1}^{K} \xi_k(t)b_k = \boldsymbol{\xi(t)'b}, \tag{2.8}$$

Plugging (2.7) and (2.8) in to (2.6), we get

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \int \boldsymbol{\xi(t)'b\phi(t)'c}_i dt, \tag{2.9}$$

Therefore, we can rewrite the link function as,

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \beta_1' c_i, \tag{2.10}$$

where $\beta_1' = \int \xi(t)'b\phi(t)'dt$, i indicates the observations; t indicates the time values; ϕ and ξ represent the spline basis functions for the data and β, respectively; c represents

the spline coefficients for the data; b represents the spline coefficients for β; and L & K are the number of basis functions used for the data and β, respectively. Again, the model can be extended to include additional covariates by updating equation 2.10,

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \beta_1' c_i + \sum_{j=2}^{p} \beta_j Z_j, \tag{2.11}$$

where, Z_j contain the data for each of our additional covariates [7].

2.2.3 Generalized Linear Models using Functional Principal Component Analysis

In the previous section we approximated the functional covariates and regression coefficients using B-spline basis approximation. Here we consider a special basis - functional principal components. Principal component analysis (PCA) can be described as the search for a probe, ξ, that captures the greatest variation in the data. In other words, we try to find ξ such that the probe scores have the largest variation,

$$\rho_\xi(x_i) = \int \xi(t) x_i(t) dt, \tag{2.12}$$

where,

$$\int \xi^2(t) dt = 1. \tag{2.13}$$

In multivariate PCA, all pairs of eigenvalues and eigenvectors are computed by solving the equation,

$$V\xi_j = \mu_j \xi_j, \tag{2.14}$$

where, V is the covariance matrix and

$$\mu_j = \sum_i \rho_\xi^2(x_i), \tag{2.15}$$

subject to $\int \xi^2(t) dt = 1$. The process is very similar in the functional setting. Here, the eigenfunctions (or harmonics), are calculated as solutions to,

$$\int v(s,t)\xi_j(t) dt = \mu_j \xi_j(s), \tag{2.16}$$

where, v(s, t) is the bivariate covariance function [11]. The function pca.fd in the fda package in R can be used to perform FPCA on a functional object. Once FPCA has been performed, the eigenvalues, μ_j, can be plotted against their indices, j, creating a scree plot which can be used to determine the number of harmonics to use. The number of harmonics used is chosen by identifying where the line starts to level-out or straighten. The point where this occurs is a visual indication of the number of harmonics that should be included. Once the number of harmonics to use has been determined, we then regress the categorical outcome variable onto the scores obtained from the FPCA using a generalized linear model with the logit link function. The model now becomes,

$$p(y_i; \theta_i; \phi) = exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\alpha(\phi)} + c(y+i, \phi)\right\}, \tag{2.17}$$

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \sum_{l=1}^{L}\gamma_{il}\beta_l, \tag{2.18}$$

where γ_{il} are the principal component scores for i individuals and l harmonics. Since $\gamma_{il} = \int \xi_l(t)(x_i(t) - \bar{x}(t))dt$, equation 2.21 now becomes,

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \int\sum_{l=1}^{L}\beta_l\xi_l(t)(x_{il}(t) - \bar{x}(t))dt, \tag{2.19}$$

giving us,

$$\beta(t) = \sum_{l=1}^{L}\beta_l\xi_l(t). \tag{2.20}$$

Incorporation of additional covariates into the model is an easy extension. Equation (2.21) now becomes,

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \beta_0 + \sum_{l=1}^{L}\gamma_{il}\beta_l + \sum_{j=L+1}^{P}\beta_j Z_j \tag{2.21}$$

where, Z_j contain the data for each of our additional covariates. Since normally only the first few principal components are needed to capture the majority of the variation within the data, FGLM using FPCA allows for dimension reduction which decreases

the degrees of freedom for error in the model. This decrease allows for a more stable estimate and, thus, may be more ideal than the standard FGLM described previously [8, 11].

2.3 Results

2.3.1 Functional linear model with thermogram data as the response and disease status as the predictor

In the Lupus thermogram framework, the response variable of interest is thermogram shape and structure predicted by disease status (case or control). Therefore, in equation 2.5, $i = 1, 2, ..., 592$; $j = 1, 2$ indicating disease status; $t =$ time points (termperature in this case); $K = 35$; and the values of x_{ij} are 0 or 1 indicating either case or control, respectively. Our design matrix is then a 592×3 matrix with the first column being all 1's, the second column cotains 1's for cases and 0's for controls, and the third column contains 1's for controls and 0's for cases.

Since we used 35 B-spline basis functions, we have 35 terms for each of the three coefficients - intercept, cases, and controls. These β values can then be plotted against temperature (a sequence that ranges from 45 to 90°C). Since there are only a few values for each coefficient the plots will look very rough. Therefore, we implement some smoothing to yield more interpretable plots. These plots give the mean thermogram (intercept), and the perturbations of the overall mean required to fit a curve for cases and controls. We can also use the predicted response values, returned to us from the regression, to get the predicted curves for both cases and controls (Figure 2.1).

Plotting the results of the functional t-test we see that the most significant differences between the curve for cases and the curve for controls lies in the $[60, 69]$°C and $[72, 85]$°C ranges (Figure 2.2). Figure 2.2 also shows the maximum value of the

test statistic (highest value of the red line), the critical value for each individual t-test performed (dotted blue line), and the overall critical value (dashed blue line). From this, the value of the test statistic, T_{obs}, was 14.22, and the critical value from the permutation test was 2.92. This indicates a significant overall difference between the curves for cases and the curves for controls.

With additional covariates

Now, we extend the above model to include additional covariates. In the thermogram application we include sex, race, and year of birth as covariates in the model. Disease status (case or control) will require one coefficient; sex (male or female) will require one additional coefficient; race (Black or White) will require one additional coefficient; and year of birth (1924-1944, 1945-1955, 1956-1971, or 1972-1993) will require three additional coefficients, making p = 7 in equation 2.5.

Figure 2.3 shows the estimated regression coefficients from this model. Now, w ith more than two groups in the model, we no longer perform the functional t-test but can, instead, implement the functional F-test. Just as with the functional t-test, we can get a plot for the funcitonal F-test results (Figure 2.4), and calculate the F-statistic = $max(F(t))$. Again, we use a permutation test to determine the critical value for performing the hypothesis test. Figure 2.4 indicates that the strongest predictive relationship between the covariates and thermogram structure lies within the $[60, 85]^{\circ}$C range. The test yields an observed statistic, F_{obs}, of 0.48 and a critical value of 0.06 indicating a strong predictive relationship between the covariates and the response variable.

The results of the linear regression with functional response variable and scalar/categorical covariate(s) shows a significant difference between curves for cases and curves for controls as well as a strong predictive relationship between disease status + covariates and thermogram structure. Both models show the largest differences

and strongest relationships occur between 60°C and 85°C. Therefore, we choose to only include DSC data within that temperature range when running the rest of the regression models.

2.3.2 Generalized linear models: disease status as the response and thermogram data as the predictor

Now we shift the focus to the case where the response variable of interest is categorical and at least one of the covariates is functional. In the Lupus data, our response variable is disease status, 1 indicating cases and 0 indicating control. The thermogram functional object now becomes the predictor variable. We want to investigate how well thermogram shape and structure predicts disease status. We run the regression using the generalized linear model. In this example we set up 15 spline basis functions (K=15) for the regression coefficients, and use 20 spline basis functions (L=20) for the data. We then use the fregre.glm function in the fda.usc package in R to run the regression. Since we use a set of basis functions for the regression coefficient basis expansion, the results from the regression yield estimated regression coefficients that are a functional data object. These are plotted in Figure 2.5 and we see that there seems to be a significant edge effect near the 85°C mark.

A χ^2 goodness-of-fit test was used to test for lack of fit. With a test-statistic of 796.50 on 576 degrees of freedom, we get a p-value ≈ 0 indicating evidence of a lack of fit in the model. Next, we split the data into a training data set and a test data set in order to evaluate the predictability of the model. A 2/3 vs 1/3 split is used giving 200 cases and 200 controls in the training set; 98 cases and 94 controls in the test set. We run the regression using only the training set and use the results to predict the response values for the test set. An observation within the test set is classified as a case if their predicted value is greater than 0.5; classified as a control if their predicted response value is less than 0.5. Comparing predicted classification

to true classification, this model yields a 76.56% correct classification rate.

With additional covariates

Motivated by the lack of fit from the previous model, we investigate the effect additional covariates have on the model. We use the same covariates as above. From the regression we get the estimated regression coefficients, test for lack of fit, and split the data into training and test sets to estimate predictability of the model. Figure 2.6 plots the regression coefficient functional data object associated with the thermogram predictor variable. Again we see a large edge effect near the 85°C mark, and we also see a big peak around 67°C that was not present in the reduced model. Summary of the regression indicates that none of the additional covariates are significant in the model. Since individuals were matched based on age and gender, these results make sense. The χ^2 goodness-of-fit test produced a test-statistic = 576.32 on 567 degrees of freedom giving a p-value = 0.38. Therefore, the addition of these covariates improves the fit of the model. After splitting the data into training and test sets, running the regression on the training set, and using this to predict the response for the test set, responses are classified into case and control if their predicted response is greater than 0.5 or less than 0.5, respectively. Comparing predicted classification to true classification this model had a 74.47% correct classification rate; a rate slightly lower than in the model with only the functional predictor variable.

Testing potential interactions

Lastly, we extend the model to test potential interactions between the scalar/categorical covariates. First, we included all potential interactions to see their effect on the model. Figure 2.7 shows the functional regression coefficients and Table 2.1 gives the estimated regression coefficients for the main effects and interaction effects for each of the additional covariates. This model yields a deviance value of 564.81 on 560 degrees

of freedom with a p-value of 0.44 indicating no significant lack of fit. From Table 1 we see that sex does not have a significant effect on the model but race, year of birth, and their interaction do. Therefore, we update the model and remove the main effects and interactions involving sex. These regression coefficients are presented in Table 2.2. This reduced model gives a deviance of 567.14 on 565 degrees of freedom and a p-value of 0.47. It also has a 74.47% correct classification rate.

2.3.3 Generalized linear models using functional principal component scores as the predictor and disease status as response

The final model we consider uses disease status, case (1) or control (0), as the response variable, and principal component scores as the functional covariate. We first perform FPCA on the thermogram data. Figure 2.8 shows the scree plot of the first 15 principal components (PCs). From the scree plot, we conclude that only the first six PCs are needed since together they explain 99% of the variation in the data (Figure 2.8). Figure 2.9 plots the overall mean thermogram curve as well as two additional lines for each PC. These two additional lines show what happens to the mean curve when a small amount of the PC is added (+) or subtracted (-). We see that the first harmonic captures 68.4% of the total variation about the mean and shows the contrast between cases and controls. The second harmonic, explaining an additional 14.5% of variation, indicates a vertical shift in the mean. The third harmonic captures the vertical shifts about the two main peaks, and the remaining harmonics capture much smaller noise and variation.

Now that we have determined the number of PCs to include, we use the function glm in the stats package in R to fit the model in Section 2.2.3. Table 2.3 gives the values, standard error, t-statistic, and p-value for the coefficient estimates and Figure 2.10 plots the functional object representing the regression coefficient estimates. We no longer see the edge effect we saw in previous models, and the estimated coeffi-

cients have a smaller magnitude than in the previous models. We get a test-statistic of 613.54 on 585 degrees of freedom when testing for a lack of fit. This gives a p-value of 0.20 indicating no evidence of a lack of fit in the model. Finally, individuals are classified as case or control if their predicted response value is more than 0.5 or less than 0.5, respectively. Comparing these predicted classifications to true disease status we get 81.25% correct classification from the model, a value greater than that from any of the previous models.

Figure 2.11 shows the first six principal components. The first principal component curve models individuals starting with excess specific heat capacity values around average that then drop below average, and then increase to above average values, eventually decreasing back to average. Thus individuals with thermograms matching this pattern will have a positive γ_i value, and individuals experiencing the opposite of this will have a negative γ_i value. This curve very closely resembles the regression curve for cases in Figure 2.1, therefore we can conclude that cases will tend to have positive γ_i values and controls will likely have negative γ_i values. From Table 2.3, we see that the first principal component is highly significant for disease status and that individuals with curves described as above have an odds ratio of $e^{6.600} = 735.10$ of being a case relative to an average person. The second principal component was also found to be significant and represents individuals that have above average excess specific heat capacity values between 55°C and 85°C. Individuals experiencing this type of vertical shift from the average curve have an odds ratio of $e^{-4.467} = 0.011$ of being a case relative to the average person.

The third principal component was not significant for disease status and the remaining three principal components explain only a small portion of the total variance. However, principal components 5 and 6 were found to be highly significant. This suggests that we may be underestimating the standard errors in the first few principal components, and this is being captured in the latter principal components.

Finally, the intercept coefficient, $\beta_0 = 0.068$ gives the probability of an average individual within our dataset having Lupus $= 1 - \frac{1}{1+e^{0.068}} = 0.52$. Given that 298 out of 592 (50.3%) individuals within the Lupus dataset truly do have lupus, the estimated beta makes sense.

2.4 Tables and Figures

Table 2.1: Estimated regression coefficients for the additional covariates and their interactions in the FGLM

Coefficient	**Estimate**	**Std. Err**	**Z value**	**p-value**
Intercept	1.20	0.47	2.58	0.01
Sex-Male	0.18	0.71	0.25	0.81
Race-White	-1.22	0.51	-2.38	0.02
Year of Birth (1944, 1955]	-0.69	0.53	-1.30	0.19
Year of Birth (1955, 1971]	-1.67	0.54	-3.11	0.002
Year of Birth (1971, 1993]	-1.31	0.58	-2.25	0.02
Sex*Race	-0.12	0.63	-0.18	0.85
RaceWhite*Year of Birth (1944, 1955]	0.87	0.74	1.18	0.24
RaceWhite*Year of Birth (1955, 1971]	2.02	0.99	2.05	0.04
RaceWhite*Year of Birth (1971, 1993]	1.13	0.66	1.71	0.09
SexMale*Year of Birth (1944, 1955]	-0.13	0.78	-0.17	0.87
SexMale*Year of Birth (1955, 1971]	0.69	0.89	0.77	0.44
SexMale*Year of Birth (1971, 1993]	0.14	0.71	0.20	0.84

Table 2.2: Estimated regression coefficients after removing sex and its interactions from the model

Coefficient	Estimate	Std. Err	Z value	p-value
Intercept	1.23	0.45	2.75	0.01
Race-White	-1.25	0.50	-2.50	0.01
Year of Birth (1944, 1955]	-0.72	0.51	-1.41	0.16
Year of Birth (1955, 1971]	-1.54	0.51	-3.03	0.002
Year of Birth (1971, 1993]	-1.28	0.56	-2.26	0.02
RaceWhite*Year of Birth (1944, 1955]	0.85	0.67	1.27	0.20
RaceWhite*Year of Birth (1955, 1971]	2.59	0.79	3.29	0.001
RaceWhite*Year of Birth (1971, 1993]	1.14	0.66	1.73	0.08

Table 2.3: Estimated regression coefficients for the first 6 principal components in the FGLM model using FPC scores

Coefficient	Estimate	Std. Err	t-value	p-value
Intercept	0.07	0.10	0.68	0.50
PC 1	6.6	0.62	10.69	< 0.001
PC 2	-4.47	1.15	-3.88	< 0.001
PC 3	-1.25	1.45	-0.87	0.39
PC 4	0.96	2.32	0.42	0.68
PC 5	18.93	3.57	5.31	< 0.001
PC 6	-17.52	5.14	-3.41	0.001

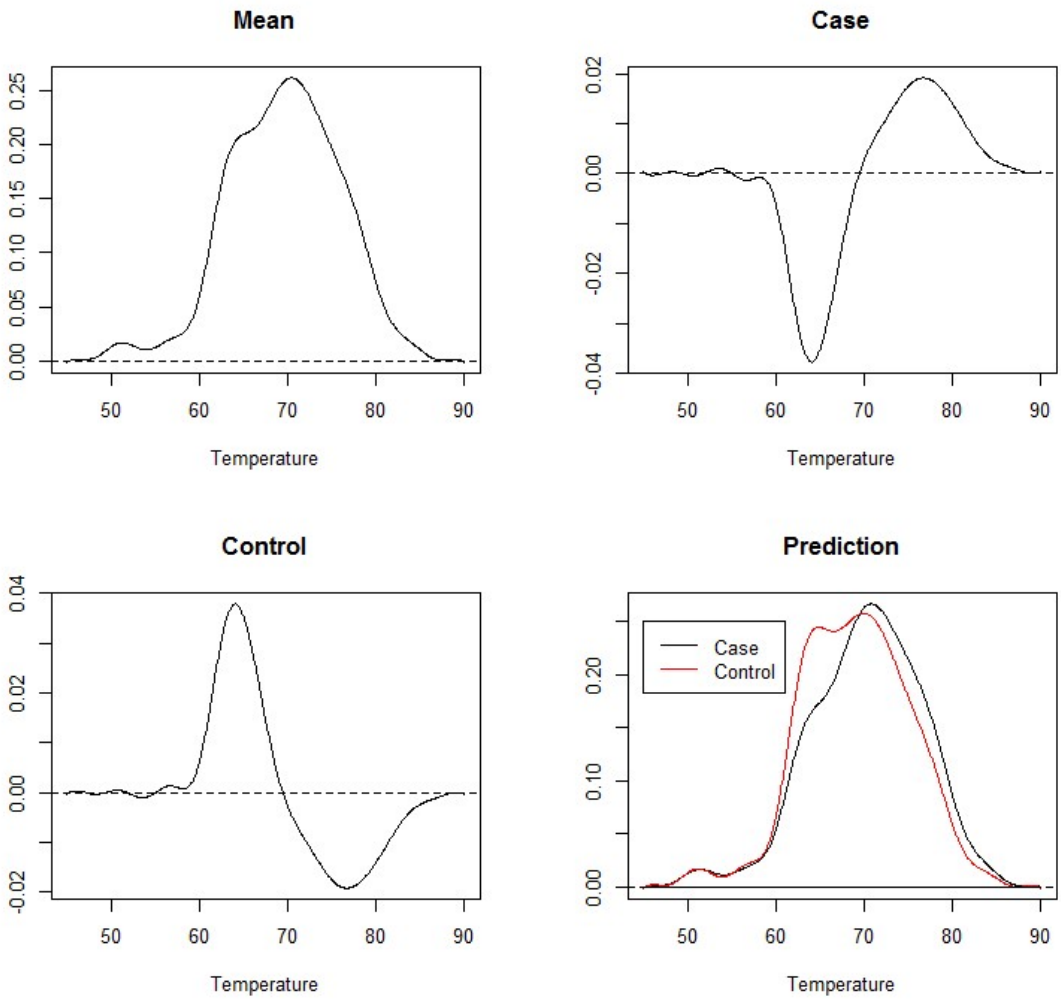

Figure 2.1: Smoothed regression coefficients estimated for predicting thermograms from disease status. The first panel is the intercept coefficient, corresponding to the overall mean thermogram. The second and third panels show the estimated perturbation (regression coefficients) of the overall mean needed to fit a curve for cases and controls respectively. The last panel shows the predicted mean thermograms for cases and controls.

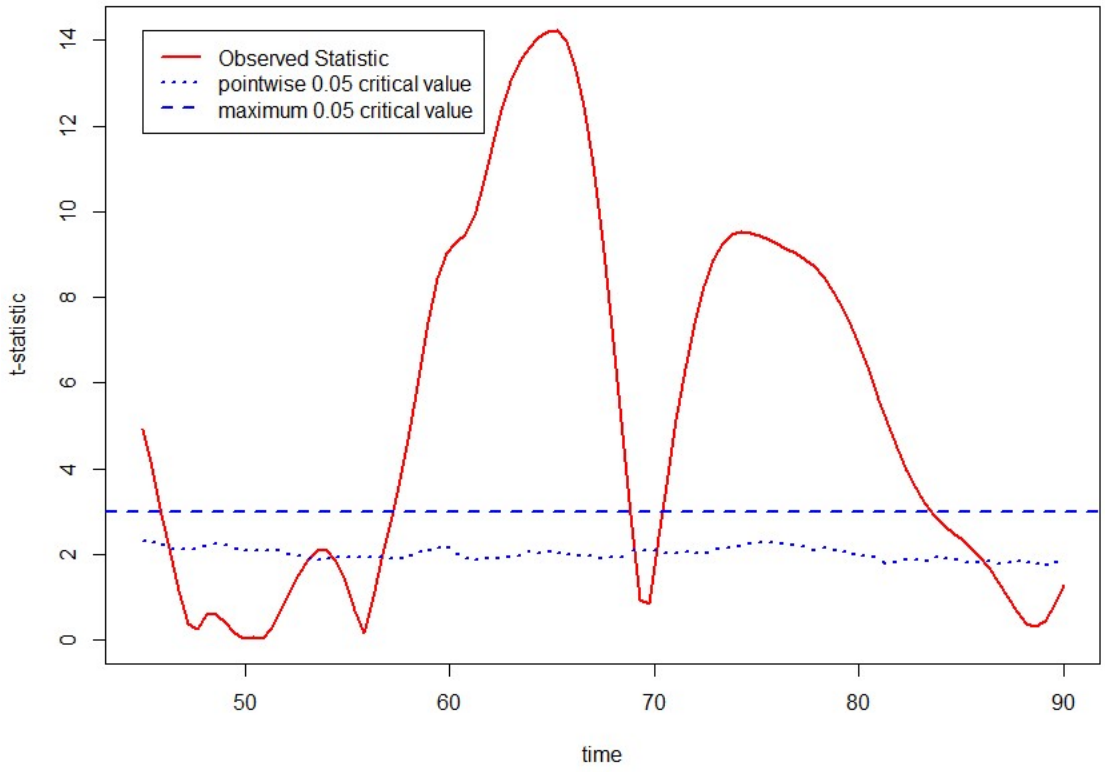

Figure 2.2: A test for the difference in thermogram profiles for cases and controls.

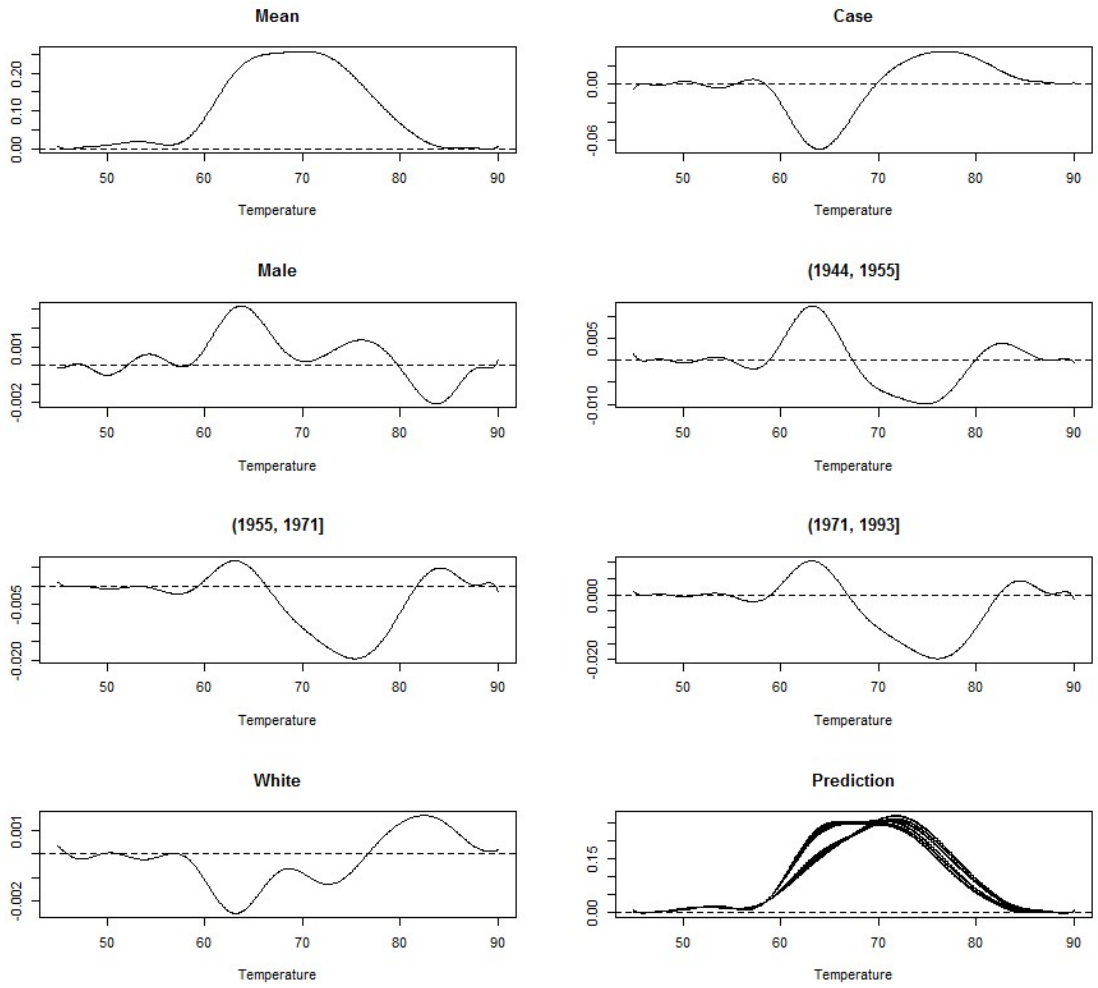

Figure 2.3: Smoothed regression coefficients estimated for predicting thermograms from disease status, sex, race, and year of birth. From left to right, top to bottom, panel 1 is the intercept coefficient, corresponding to the overall mean thermogram. The second panel shows the estimated perturbation (regression coefficients) of the overall mean needed to fit a curve for cases. The third panel shows the estimated perturbation of the overall mean needed to fit a curve for males. Panels 4-6 show the estimated perturbation of the overall mean thermogram needed to fit a curve for individuals with a birth year in (1944, 1995], (1955, 1971], and (1971, 1993], respectively. Panel 7 shows the estimated perturbation of the overall mean thermogram needed to fit a curve for individuals identifying as White. Finally, panel 8 shows the predicted curves for each combination of disease status, sex, year of birth, and race.

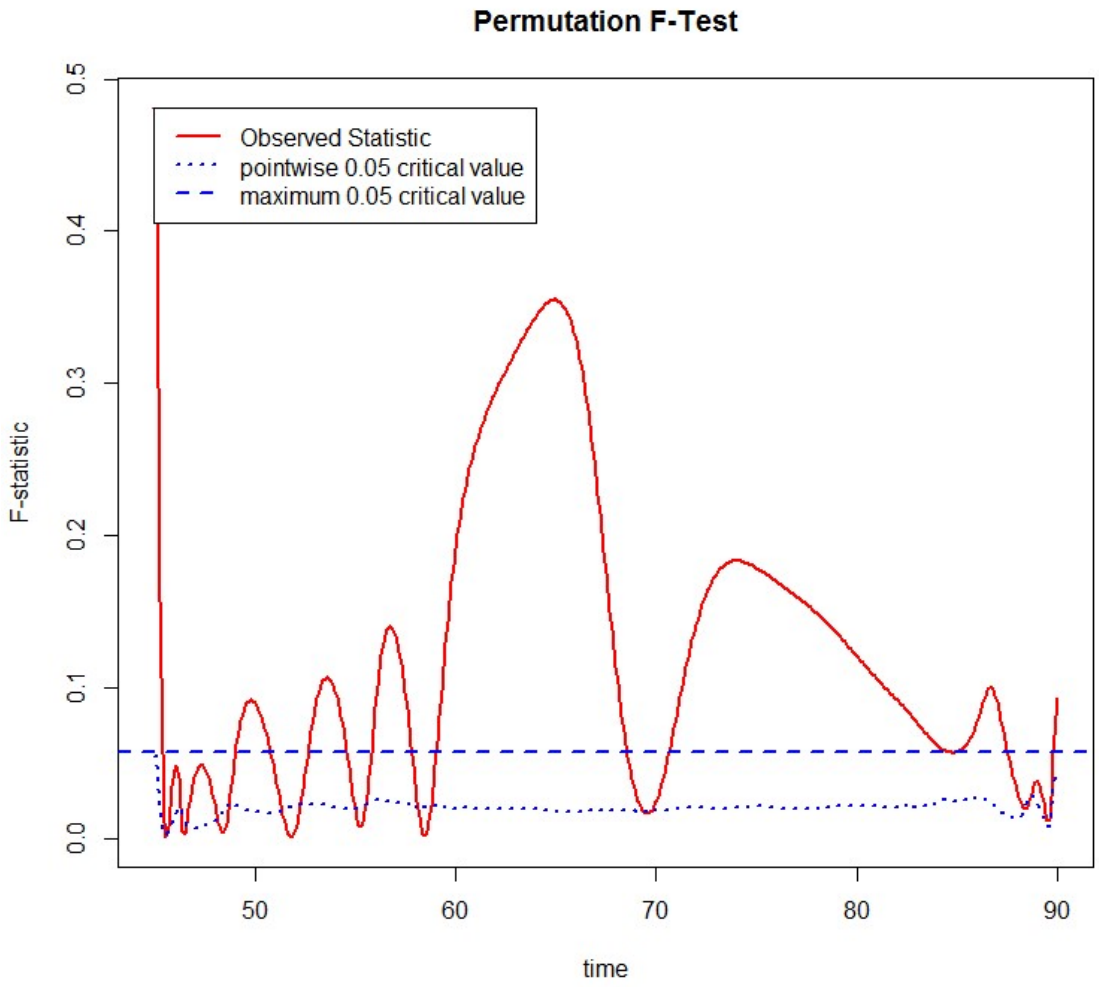

Figure 2.4: Permutation test for a predictive relationship between disease status, sex, race, and year of birth and thermogram structure.

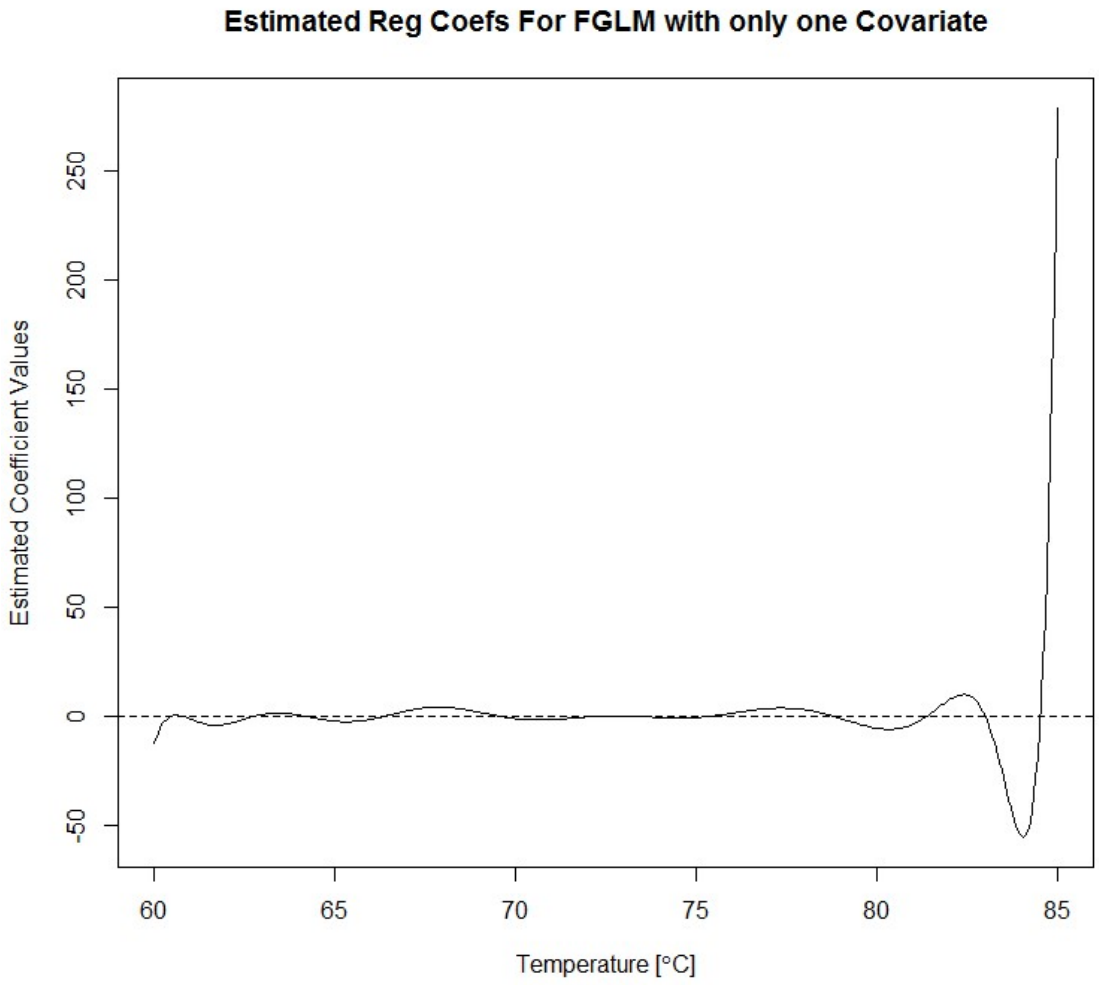

Figure 2.5: Estimated regression coefficients for the FGLM when disease status is the response variable with thermograms as the functional predictor variable.

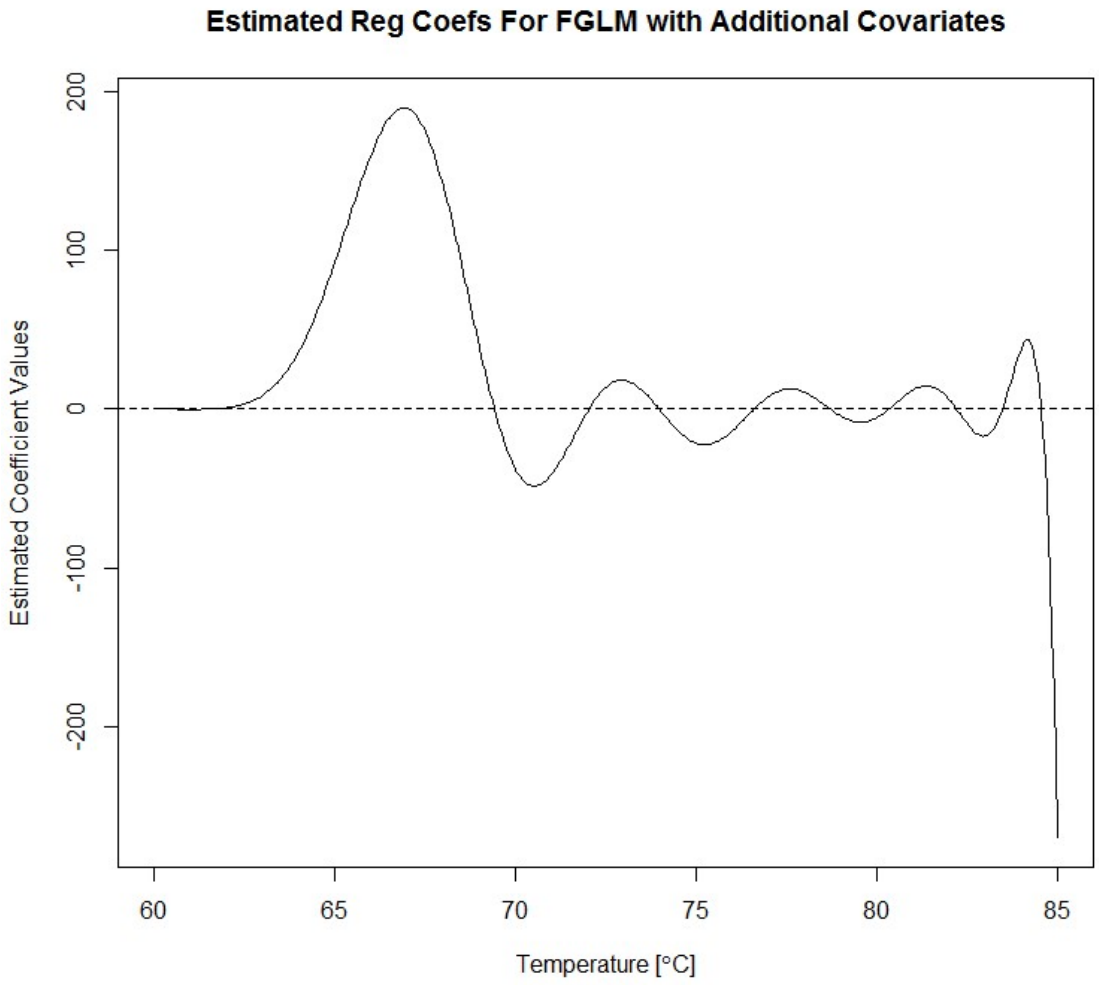

Figure 2.6: Estimated regression coefficients for the thermogram predictor variable in the FLGM framework when additional covariates are added to the model.

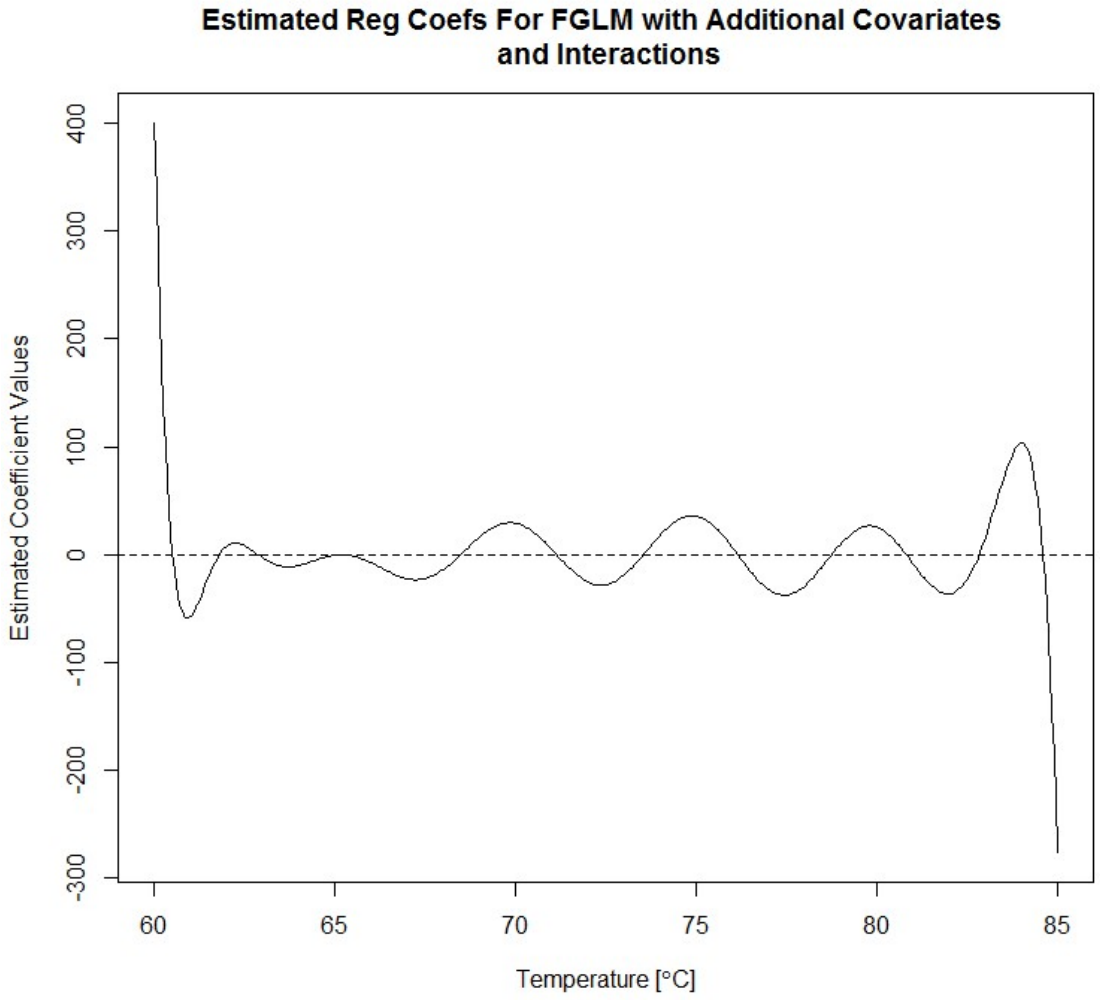

Figure 2.7: Estimated regression coefficients for the thermogram predictor variable in the FLGM framework when additional covariates and their interactions are added to the model.

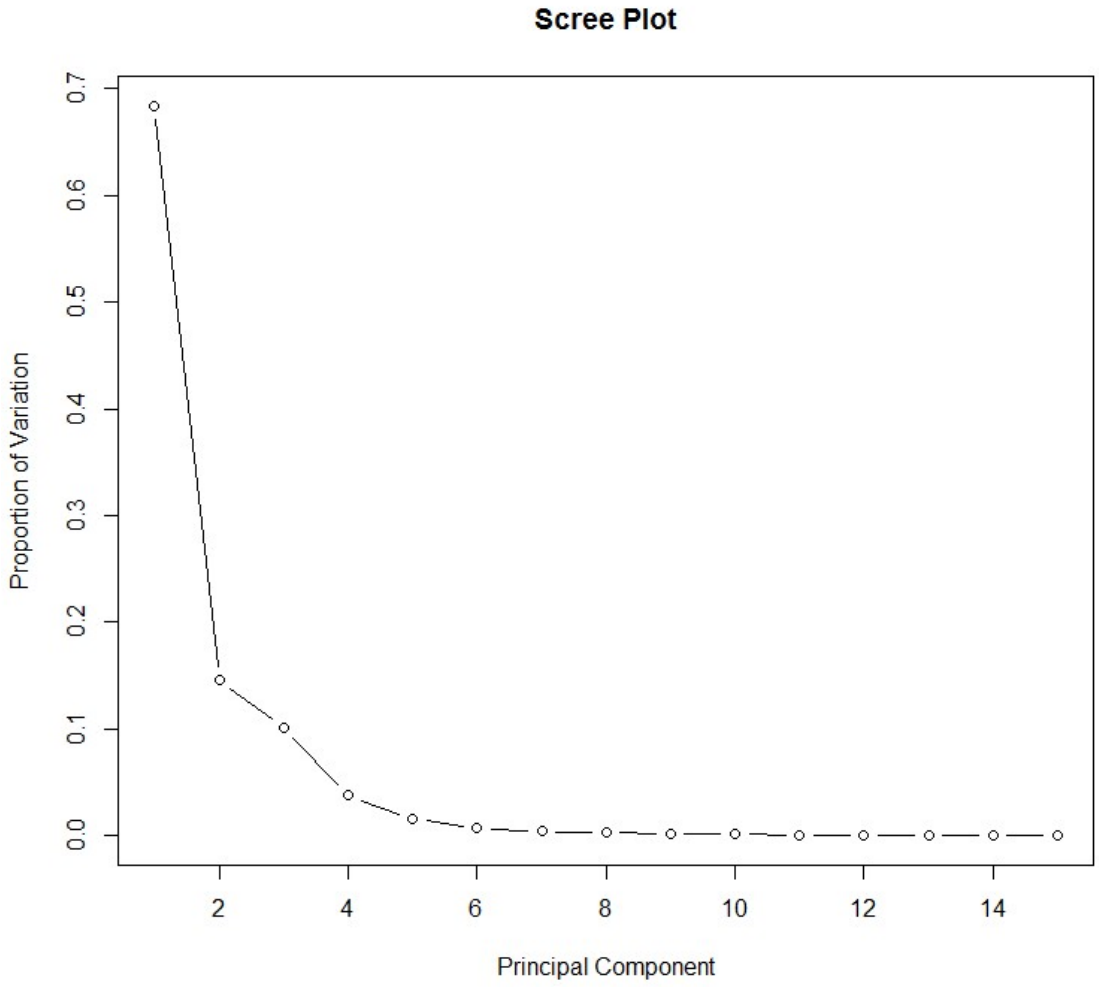

Figure 2.8: Scree plot of the first 15 PC's resulting from FPCA on thermogram data

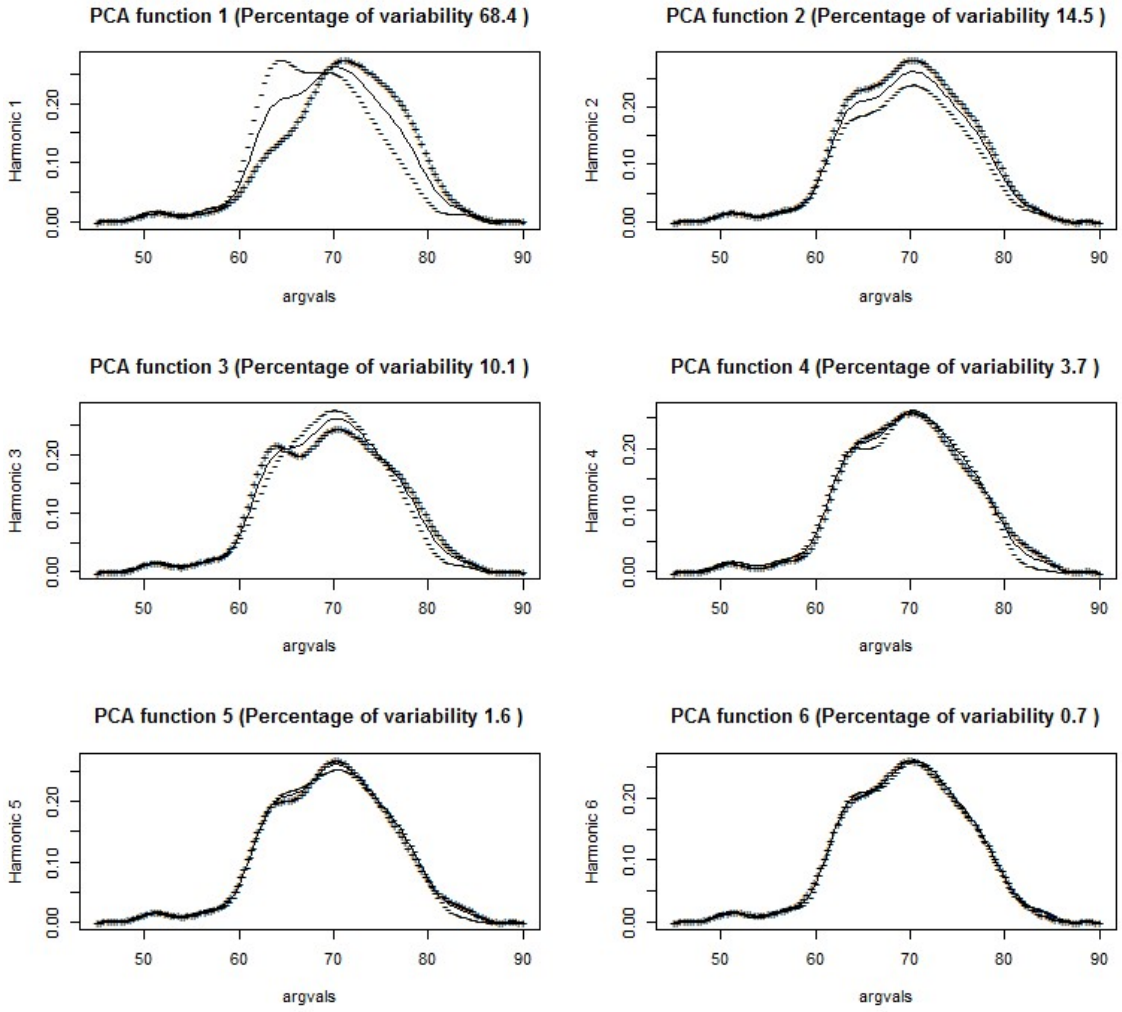

Figure 2.9: The first 6 functional PC's and the percent variation they capture.

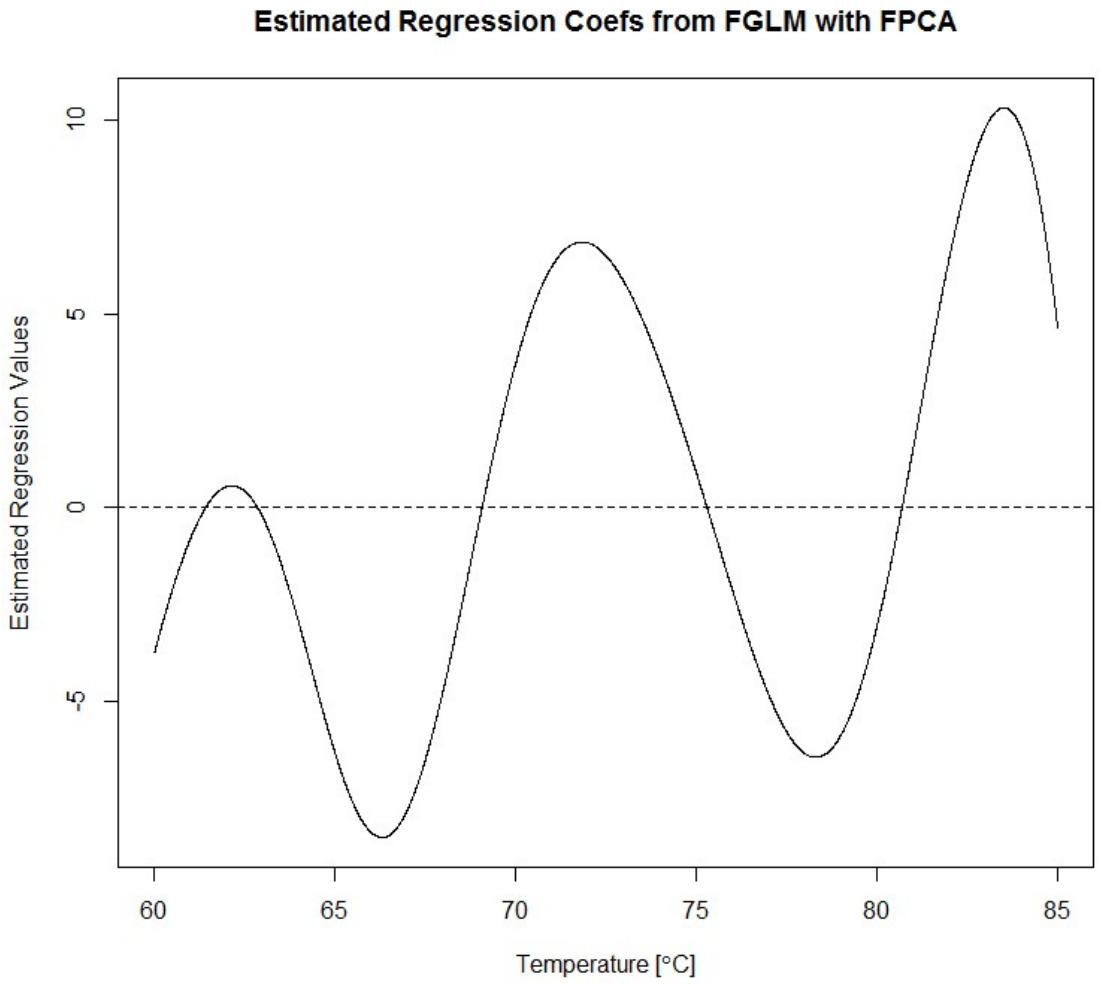

Figure 2.10: Estimated regression coefficients associated with the thermogram predictor variable when disease status is the response variable and principal component scores from FPCA are the predictor variables.

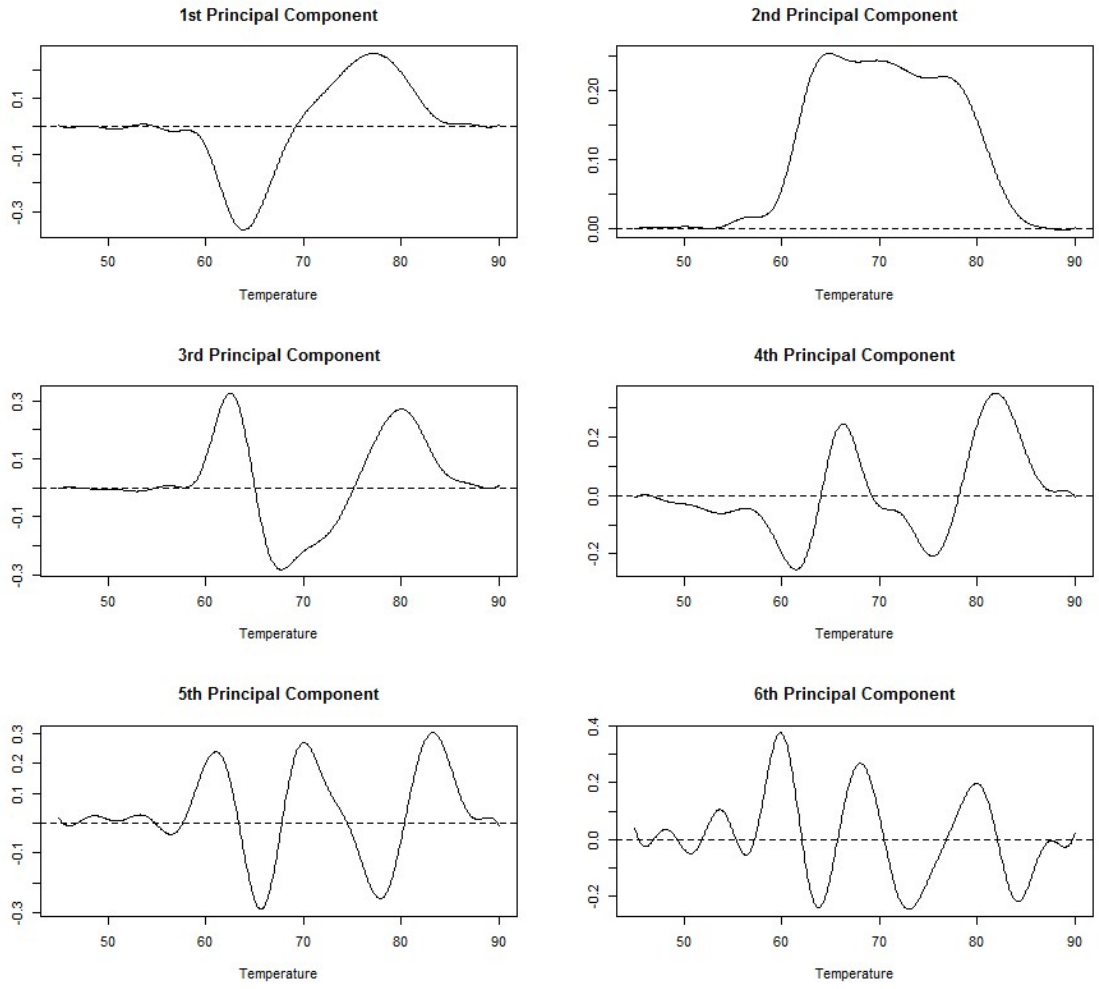

Figure 2.11: The first 6 principal component curves for thermogram profiles.

CHAPTER 3

FUNCTIONAL GENERALIZED PARTIALLY LINEAR SINGLE-INDEX MODEL

3.1 Introduction

In project 1, we assumed the relationship between the logit of the response variable and the predictor variables is linear. However, this may not always be the case; there may be some curvature in the logit. A violation of the linearity assumption can give misleading results. An alternative to the standard linear-logistic regression model is the generalized single-index model, $g(E(Y)) = \eta(x'\beta)$, where η is an unknown function. Along with being able to capture any non-linearity in the logit, single-index models also provide increased flexibility, circumvent the curse of dimensionality, and yield interpretable results.

In our second project, we address the restrictiveness of the parametric model with our proposed model: generalized functional partially linear single-index model (GFPL). Shujie Ma proposed the functional single index model (FSiM) in 2014. Ma used B-spline basis functions to approximate the slope and link function and implemented an iterative estimating method using the least squares criterion to estimate the functions [12]. Carroll et al. developed the generalized partially linear single index model (GPLSiM) in 1997 [13]. The GPLSIM is useful when the relationship may not be linear. Carroll et al. implements an unknown nonparametric function to allow for nonlinearity in the logit and implement an iterative estimation procedure to estimate the slope and the link function. Ma's method can not handle categorical

responses. On the other hand, Carrol et al. proposed a method that is not applicable to functional data. Therefore, we propose the GFPL which borrows from both of these proposed methods to build our new method.

3.2 Methods

3.2.1 The Model

Let Y denote the response variable, which belongs to a *natural exponential family* with *natural parameter* θ. That is, Y has the density

$$f(y|\theta) = c(y)\exp(\theta y - b(\theta)), \tag{3.1}$$

with respect to a σ-finite measure ν. Let Θ be the *natural parameter space*, i.e. the set of all θ such that $0 < \int c(y)\exp(\theta y)d\nu < \infty$. Then Θ is convex, and all derivatives of $b(\theta)$ and all moments of y exist for $\theta \in \Theta^0$, the interior of Θ. Denote $E_\theta[Y] = b'(\theta)$ by $\mu(\theta)$ and $\text{var}_\theta(Y) = b''(\theta)$ by $\sigma(\theta)$.

Let $\mathbf{X}(t) = \left(X^{(1)}(t), \cdots, X^{(p)}(t)\right)^T$ and $\mathbf{Z}(t) = \left(Z^{(1)}(t), \cdots, Z^{(q)}(t)\right)^T$ denote p-dimensional and q-dimensional functional predictors with a compact support $\mathcal{T}$, respectively. A generalized functional model satisfies

(i) The functional covariates $\mathbf{X}(t)$ and $\mathbf{Z}(t)$ influence Y in the form of a linear combination

$$\begin{aligned}\gamma = & \sum_{j=1}^{p}\int_{\mathcal{T}}\alpha_{j,0}(t)X^{(j)}(t)dt + \sum_{k=1}^{q}\int_{\mathcal{T}}\beta_{k,0}(t)Z^{(k)}(t)dt \\ = & \int_T \boldsymbol{\alpha}_0^T(t)\mathbf{X}(t)dt + \int_T \boldsymbol{\beta}_0^T(t)\mathbf{Z}(t)dt,\end{aligned} \tag{3.2}$$

where $\boldsymbol{\alpha}_0(t) = (\alpha_{1,0}(t), \cdots, \alpha_{p,0}(t))^T$ and $\boldsymbol{\beta}_0(t) = (\beta_{1,0}(t), \cdots, \beta_{q,0}(t))^T$ are p-dimensional and q-dimensional smooth coefficient functions.

(ii) The influence is related to $\mu(\theta)$, the mean of Y, by a known injective link

function $g : \mathbb{R} \to \mathbb{R}$: $\gamma = g(\mu(\theta))$. The link function g is defined as in [14]. We also use the injective function $h = (g \circ \mu)^{-1}$ to relate γ to the natural parameter θ, i.e. $\theta = h(\gamma)$.

As pointed out by [13], the linear structure in Equation 3.2 is not always complex enough to model the relationship between the response variable and the associated covariates. We consider the following generalized functional partially linear single index model:

$$g\left(\mu\left(\theta\right)\right) = \eta_0\left(\int_{\mathcal{T}} \boldsymbol{\alpha}_0^T(t)\mathbf{X}(t)dt\right) + \int_{\mathcal{T}} \boldsymbol{\beta}_0^T(t)\mathbf{Z}(t)dt, \tag{3.3}$$

where η_0 is an unknown nonconstant smooth function.

3.2.2 Notations and regular conditions

To develop our GFPL model, we first introduce some notations as follows. Given a random sample $Z_1, \cdots, Z_n$, let $\mathbb{G}_n(f) = \mathbb{G}_n(f(Z_i)) := n^{-1/2}\sum_{i=1}^n (f(Z_i) - E[f(Z_i)])$ and $\mathbb{E}_n f = \mathbb{E}_n f(Z_i) := \sum_{i=1}^n f(Z_i)/n$. We use $\|\cdot\|_r$ to denote the l_r-norm of a vector or function. In particular, we denote the l_2-norm by $\|\cdot\|$. Given any function f, $\|f\|_{r,A}$ denotes the l_r-norm of f on the area $\mathcal{A}$. Let $\wedge$ and $\vee$ denote the minimum operator and the maximum operator respectively. Given square matrices $\mathbf{A}$ and $\mathbf{B}$, we write $A \leq B$ if and only if $B - A$ is non-negative definite, and we use $\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ to denote the minimum and the maximum eigenvalue of the matrix.

(C1) $\alpha_{j,0}(\cdot) \in \mathcal{C}^{(d)}(\mathcal{T}), \beta_{k,0}(\cdot) \in \mathcal{C}^{(d)}(\mathcal{T}), \forall 1 \leq j \leq p, 1 \leq k \leq q$, and $\eta_0(\cdot) \in \mathcal{C}^{(d)}(\mathcal{U})$, where $\mathcal{C}^{(d)}(\mathcal{Y}) := \left\{\phi : \left\|\phi^{(d)}\right\|_{\infty,\mathcal{Y}} \leq C\right\}$ denotes the space of functions with the dth order derivative bounded by C over a given set $\mathcal{Y}$, for some integer $d \geq 2$ and constant C.

(C2) The functionals $X^{(j)}(\cdot); Z^{(k)}(\cdot); j = 1, \cdots, p; k = 1, \cdots, q$, are random variables

on $\mathcal{X}(\mathcal{T})$, where

$$\mathcal{X}(\mathcal{T}) := \left\{ \phi : \max_{1\le j\le p} \sup_{t_1\neq t_2, t_1,t_2\in\mathcal{T}} \frac{|\phi(t_1)-\phi(t_2)|}{|t_1-t_2|} \le C \right\}.$$

(C3) The function $h(\cdot)$ and $b(\cdot)$ have three continuous derivatives with $\|h'''(\cdot)\|_{\infty,\mathcal{U}}$, $\|b'''(\cdot)\|_{\infty,\mathcal{U}} \le C_D$.

(C4) Let $m_{*n} = \min_{1\le i\le n} m_i$ and $m_n^* = \max_{1\le i\le n} m_i$.

$$\max_{1\le i\le n} \frac{\max_{1\le l\le m_i} t_{i,l+1} - t_{i,l}}{\min_{1\le l\le m_i} t_{i,l+1} - t_{i,l}} \le C_K,$$

for some constant $C_k > 0$.

Remark 3.2.1. Conditions (C1) and (C2) are widely adopted in functional analysis literature (see [12,15]). From the compactness of $\mathcal{T}$ and Conditions (C1) and (C2), it can be inferred that $\int_{\mathcal{T}} \boldsymbol{\alpha}^T(t)\mathbf{X}(t)dt$, $\int_{\mathcal{T}} \boldsymbol{\beta}^T(t)\mathbf{Z}(t)dt$, and $\eta_0(\int_{\mathcal{T}} \boldsymbol{\alpha}^T(t)\mathbf{X}(t)dt) + \int_{\mathcal{T}} \boldsymbol{\beta}^T(t)\mathbf{Z}(t)dt$ are all bounded uniformly over $X^{(j)}(\cdot), Z^{(k)}(\cdot) \in \mathcal{X}$, $j = 1, \cdots, p, k = 1, \cdots, q$,. Thus, without loss of generality, $\mathcal{U}$ can be regarded as a bounded set in $\mathbb{R}$. The boundedness of $\mathcal{U}$, coupled with Condition (C3) implies that $h(\cdot), h'(\cdot)$ and $\sigma(\cdot)$ are bounded in $\mathcal{U}$.

Estimation

Let $\mathbf{B}_{n,1}(t) = (B_{1,1}(t), \cdots, B_{J_{n,1},1}(t))^T$ and $\mathbf{B}_{n,2}(t) = (B_{1,2}(t), \cdots, B_{J_{n,2},2}(t))^T$ be sets of κ_1th and κ_2th order normalized B-spline basis functions with knot sequences $\{\tau_s\}$ and $\{\nu_r\}$, respectively, where $\{\tau_s\}$ and $\{\nu_r\}$ satisfy $\tau_1 = \cdots = \tau_{\kappa_1} < \tau_{\kappa_1+1} < \cdots < \tau_{J_{n,1}} < \tau_{J_{n,1}+1} = \cdots = \tau_{J_{n,1}+\kappa_1}$ and $\nu_1 = \cdots = \nu_{\kappa_2} < \nu_{\kappa_2+1} < \cdots < \nu_{J_{n,2}} < \nu_{J_{n,2}+1} = \cdots = \nu_{J_{n,2}+\kappa_2}$. Following the literature of spline estimators [16, 17], we also require

$$\frac{\max_{\kappa_1\le s\le J_{n,1}} \tau_{s+1} - \tau_s}{\min_{\kappa_1\le s\le J_{n,1}} \tau_{s+1} - \tau_s} \vee \frac{\max_{\kappa_2\le r\le J_{n,2}} \nu_{r+1} - \nu_r}{\min_{\kappa_2\le r\le J_{n,2}} \nu_{r+1} - \nu_r} < C_K,$$

uniformly in n, to investigate the asymptotic properties of our proposed estimators. Throughout the rest of the chapter, we choose $\kappa_1 = \kappa_2 = \kappa$ to ease the presentation and simplify the theoretical development. However, our results can easily be extended to the $\kappa_1 \neq \kappa_2$ case. In addition, we may suppress the dependence of $J_{n,1}, J_{n,2}, \mathbf{B}_{n,1}(\cdot)$, and $\mathbf{B}_{n,2}(\cdot)$ on n for notational simplicity, when there is no confusion.

The unknown link function $\eta_0(\cdot)$ and the coefficient functions $\alpha_{j,0}(\cdot)$ and $\beta_{k,0}(\cdot)$ can be approximated by B-spline functions:

$$\eta_0(u) \approx \eta^*(u) = \sum_{s=1}^{J_1} \lambda_s^* B_{s,1}(u) = \boldsymbol{\lambda}^{*T} \mathbf{B}_1(u),$$

$$\alpha_{j,0}(t) \approx \alpha_j^*(t) = \sum_{r=1}^{J_2} \delta_{r,j}^* B_{r,2}(t) = \boldsymbol{\delta}_j^{*T} \mathbf{B}_2(t),$$

$$\text{and} \quad \beta_{k,0}(t) \approx \beta_{k,0}^*(t) = \sum_{r=1}^{J_2} \omega_{r,k}^* B_{r,2}(t) = \boldsymbol{\omega}_k^{*T} \mathbf{B}_2(t),$$

where $\boldsymbol{\lambda}^* = (\lambda_1, \cdots, \lambda_{J_1})^T$, $\boldsymbol{\delta}_j^* = (\delta_{1,j}, \cdots, \delta_{J_2,j})^T, j = 1, \cdots, p$, and $\boldsymbol{\omega}_k^* = (\omega_{1,k}, \cdots, \omega_{J_2,k})^T, k = 1, \cdots, q$, such that

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \|\eta_0(u) - \boldsymbol{\lambda}^T \mathbf{B}_1(u)\|_\infty, \quad \boldsymbol{\delta}_j^* = \arg\min_{\boldsymbol{\delta}_j} \|\alpha_{j,0}(t) - \boldsymbol{\delta}_j^T \mathbf{B}_2(t)\|_\infty, \quad 1 \leq j \leq p,$$

$$\text{and} \quad \boldsymbol{\omega}_k^* = \arg\min_{\boldsymbol{\omega}_k} \|\beta_{k,0}(t) - \boldsymbol{\omega}_k^T \mathbf{B}_2(t)\|_\infty, \quad 1 \leq k \leq q.$$

According to [18], under Condition (C1)

$$\sup_{u \in \mathcal{I}} |\eta^*(u) - \eta_0(u)| = O(J_1^{-d}), \tag{3.4}$$

$$\sup_{t \in \mathcal{T}, 1 \leq j \leq p} |\alpha_j^*(t) - \alpha_j(t)| = O(J_2^{-d}), \quad \text{and} \tag{3.5}$$

$$\sup_{t \in \mathcal{T}, 1 \leq k \leq q} |\beta_k^*(t) - \beta_k(t)| = O(J_2^{-d}). \tag{3.6}$$

As the curves $\left\{X_i^{(j)}(t), j = 1, \cdots, p\right\}$ and $\left\{Z_i^{(k)}(t), k = 1, \cdots, q\right\}$ are discretized

at dense time points $t_{i,1}, \cdots, t_{i,m_i+1}$ with $m_i \to \infty$ as $n \to \infty$, by Riemann integration,

$$\begin{aligned}
&\int_{\mathcal{T}} \alpha_j(t) X_i^{(j)}(t) dt \\
&\qquad \approx \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) \alpha_j(t_{i,l}) X_i^{(j)}(t_{i,l}) \\
&\qquad \approx \sum_{r=1}^{J_2} \delta_{r,j} \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) B_{r,2}(t_{i,l}) X_i^{(j)}(t_{i,l})
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
&\int_{\mathcal{T}} \beta_k(t) Z_i^{(k)}(t) dt \\
&\qquad \approx \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) \beta_k(t_{i,l}) Z_i^{(k)}(t_{i,l}) \\
&\qquad \approx \sum_{r=1}^{J_2} \omega_{r,k} \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) B_{r,2}(t_{i,l}) Z_i^{(k)}(t_{i,l}).
\end{aligned} \tag{3.8}$$

We further introduce some notations as follows:

$$\begin{aligned}
&\phi_{i,r,j} = \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) B_{r,2}(t_{i,l}) X_i^{(j)}(t_{i,l}), \\
&\boldsymbol{\phi}_{i,j} = (\phi_{i,1,j}, \cdots, \phi_{i,J_2,j})^T, \boldsymbol{\phi}_i = (\boldsymbol{\phi}_{i,1}^T, \cdots, \boldsymbol{\phi}_{i,p}^T)^T, \\
&\psi_{i,r,k} = \sum_{l=1}^{m_i} (t_{i,l+1} - t_{i,l}) B_{r,2}(t_{i,l}) Z_i^{(k)}(t_{i,l}), \\
&\boldsymbol{\psi}_{i,k} = (\psi_{i,1,k}, \cdots, \psi_{i,J_2,k})^T, \boldsymbol{\psi}_i = (\boldsymbol{\psi}_{i,1}^T, \cdots, \boldsymbol{\psi}_{i,q}^T)^T, \\
&\boldsymbol{\delta} = (\boldsymbol{\delta}_1^T, \cdots, \boldsymbol{\delta}_p^T)^T, \quad \boldsymbol{\omega} = (\boldsymbol{\omega}_1^T, \cdots, \boldsymbol{\omega}_q^T)^T, \quad \text{and} \quad \boldsymbol{\zeta} = (\boldsymbol{\delta}^T, \boldsymbol{\omega}^T)^T.
\end{aligned}$$

Let

$$L_n(\boldsymbol{\lambda}, \boldsymbol{\zeta}) := \sum_{i=1}^{n} h\big(\mathbf{B}_1^T \left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}\right) \boldsymbol{\lambda} + \boldsymbol{\psi}_i^T \boldsymbol{\omega}\big) Y_i - b\left(h\left(\mathbf{B}_1^T \left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}\right) \boldsymbol{\lambda} + \boldsymbol{\psi}_i^T \boldsymbol{\omega}\right)\right) \tag{3.9}$$

be the log-likelihood function for the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$. In practice, it is almost infeasible to maximize Equation 3.9 with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$ simultaneously. Thus, we apply an iterative estimation method as is frequently used in the partially linear single index model.

Step 1: (Initialization step). Obtain an initial value $\boldsymbol{\zeta}^{init}$ by assuming a parametric form for $\eta_0(\cdot)$, and set $\boldsymbol{\delta}^{init} = \boldsymbol{\delta}^{init}\left(\sum_{j=1}^{p} \boldsymbol{\delta}_j^{init T}\left(\int_{\mathcal{T}} \mathbf{B}_2(t)\mathbf{B}_2^T(t)dt\right)\boldsymbol{\delta}_j^{init}\right)^{-1/2}$.

Step 2: Obtain $\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init}) = (\hat{\lambda}_1(\boldsymbol{\zeta}^{init}), \cdots, \hat{\lambda}_{J_1}(\boldsymbol{\zeta}^{init}))^T$ as

$$\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init}) = \arg\max_{\boldsymbol{\lambda}} \ell_n(\boldsymbol{\lambda}; \boldsymbol{\zeta}^{init}), \tag{3.10}$$

where $\ell_n(\boldsymbol{\lambda}; \boldsymbol{\zeta}^{init}) := L_n(\boldsymbol{\lambda}; \boldsymbol{\zeta}^{init})$ is constructed based on the approximation in Equation 3.7 and Equation 3.8, and $b(\cdot)$ is defined in Equation 3.1. Consequently, the estimators $\hat{\eta}(u; \boldsymbol{\zeta}^{init})$ of $\eta_0(u)$ and $\hat{\eta}'(u; \boldsymbol{\zeta}^{init})$ of $\eta_0'(u)$ can be obtained as

$$\hat{\eta}(u; \boldsymbol{\zeta}^{init}) = \mathbf{B}_1^T(u)\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init}), \quad \hat{\eta}'(u; \boldsymbol{\zeta}^{init}) = \mathbf{B}'^T_1(u)\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init}).$$

Step 3: Update $\hat{\boldsymbol{\zeta}}$ as

$$\hat{\boldsymbol{\zeta}} = \underset{\boldsymbol{\zeta}:\ \delta_{1,1} \le \cdots \le \delta_{J_2,1}}{\arg\max} \tilde{\ell}_n(\boldsymbol{\zeta}; \hat{\boldsymbol{\lambda}}), \tag{3.11}$$

where $\tilde{\ell}_n(\boldsymbol{\zeta}; \hat{\boldsymbol{\lambda}}) := L(\hat{\boldsymbol{\lambda}}; \boldsymbol{\zeta})$, and set $\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\delta}}\left(\sum_{j=1}^{p} \hat{\boldsymbol{\delta}}_j^T\left(\int_{\mathcal{T}} \mathbf{B}_2(t)\mathbf{B}_2^T(t)dt\right)\hat{\boldsymbol{\delta}}_j\right)^{-1/2}$.

Step 4: Repeat Steps 2 and 3 until convergence.

Step 5: Obtain $\hat{\boldsymbol{\lambda}}$ by using $\hat{\boldsymbol{\zeta}}$ from Step 2. The final estimate of η_0 is $\mathbf{B}_1^T(u)\hat{\boldsymbol{\lambda}}$.

Theoretical properties

Proposition 3.2.1. *Under Conditions (C1) – (C4), if* $J_1 = o\left(n^{1/3} \wedge J_2^{2d/3} \wedge m_{*n}^{2/3}\right)$, $J_2 = o(J_1^2)$, $J_1^{5/2}J_2^{1/2}(\log n)^2 = o(n)$ *, then for any* a_n *that satisfies* $n^{-1/2} = O(a_n)$ *and* $a_n = o(J_1^{-3/2})$*, we have*

(i)

$$\sup_{\|\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\|\le a_n} \left\|\hat{\eta}(u;\boldsymbol{\zeta}^{init})-\eta_0(u)\right\|_\infty = O_p\Big(\left(b_n+n^{-1/2}\log n\right)J_1^{1/2}\Big),$$

$$\sup_{\|\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\|\le a_n} \left\|\hat{\eta}'(u;\boldsymbol{\zeta}^{init})-\eta_0'(u)\right\|_\infty = O_p\Big(\left(b_n+n^{-1/2}\log n\right)J_1^{3/2}\Big),$$

where $b_n = J_1^{-d}+J_2^{-d}+m_{*n}^{-1}+a_nJ_2^{-1/2}$.

(ii) let $U_i = \int_{\mathcal{T}}\boldsymbol{\alpha}_0^T(t)\mathbf{X}_i(t)dt$, $\gamma_i = \eta_0(U_i)+\int_{\mathcal{T}}\boldsymbol{\beta}_0^T(t)\mathbf{Z}_i(t)dt$, $\rho_i = h'(\gamma_i)\Big(Y_i-\mu\left(h\left(\gamma_i\right)\right)\Big)$, $F_i = \left(h'\left(\gamma_i\right)\right)^2\sigma(h(\gamma_i)$, *and* $\mathbf{W}_i = \left(\eta_0'(U_i)\boldsymbol{\phi}_i^T,\boldsymbol{\psi}_i^T\right)^T$,

$$\left\|\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init})-\boldsymbol{\lambda}^*-n^{-1}\mathbf{V}_1^{-1}\sum_{i=1}^n\rho_i\mathbf{B}_1\left(U_i\right)+\mathbf{V}_1^{-1}E\left[F_i\mathbf{B}_1(U_i)\mathbf{W}_i{}^T\right]\left(\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right)\right\|$$
$$= O_p\left(\left(J_1^{-d}+J_2^{-d}+m_{*n}^{-1}\right)J_1^{1/2}\right)+o_p\left(J_1^{1/2}J_2^{-1/2}a_n\right).$$

where $\mathbf{V}_1 = E\left[F_i\mathbf{B}_1\left(U_i\right)\mathbf{B}_1^T\left(U_i\right)\right]$.

The result (i) in Proposition 3.2.1 establishes the uniform convergence rates of the resulting estimators $\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init})$, $\hat{\eta}(u;\boldsymbol{\zeta}^{init})$ and $\hat{\eta}'(u;\boldsymbol{\zeta}^{init})$ from **Step** 1, for any given $\boldsymbol{\zeta}^{init}$ in a a_n-neighborhood of $\boldsymbol{\zeta}^*$. The result (ii) provides a linear characterization of $\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init})$, which shows that the estimation error consists of three pieces: (1a) the systematic error, $O_p\left(\left(J_1^{-d}+J_2^{-d}+m_{*n}^{-1}\right)J_1^{1/2}\right)$, that results from the spline approximation errors in (3.4) and (3.6), and the integration approximation errors in (3.7) and (3.8); (1b) the oracle estimation error, $n^{-1}\mathbf{V}_1^{-1}\sum_{i=1}^n\rho_i\mathbf{B}_1\left(U_i\right)$, which is obtained by assuming that U_i and $\int_{\mathcal{T}}\boldsymbol{\beta}_0^T(t)\mathbf{Z}_i(t)dt$ are hypothetically known in advance; and (1c) the error introduced by the initial $\boldsymbol{\zeta}^{init}$, $n^{-1}\mathbf{V}_1^{-1}E\left[F_i\mathbf{B}_1(U_i)\mathbf{W}_i{}^T\right]\left(\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right)+o_p(J_1^{1/2}J_2^{-1/2}a_n)$.

Next, we establish the theoretical properties of the updated estimator $\hat{\boldsymbol{\zeta}}$ in **Step** 2 by using $\hat{\lambda}$ from **Step** 1.

Proposition 3.2.2. *Under the conditions in Proposition 3.2.1, we have*

(i)

$$\sup_{\|\zeta^{init}-\boldsymbol{\zeta}^*\|\leq a_n} \left\|\hat{\boldsymbol{\zeta}}-\boldsymbol{\zeta}^*\right\| = O_p\Big(\left(b_n+n^{-1/2}\log n\right)J_2^{1/2}\Big).$$

(ii)

$$\begin{aligned}\sup_{\|\zeta^{init}-\boldsymbol{\zeta}^*\|\leq a_n} \Bigg\| & \hat{\boldsymbol{\zeta}}-\boldsymbol{\zeta}^* - n^{-1}\mathbf{V}_2^{-1}\sum_{i=1}^{n}\rho_i\mathbf{W}_i + \\ & n^{-1}\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}\sum_{j=1}^{n}\rho_j\mathbf{B}_1\left(U_j\right) \\ & -\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j){\mathbf{W}_j}^T\right]\left(\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right)\Bigg\| \\ = O_p & \left(\left(J_1^{-d}+J_2^{-d}+m_{*n}^{-1}\right)J_2^{1/2}\right)+o_p\left(a_n\right).\end{aligned}$$

Similar to Proposition 3.2.1, the result (i) of Proposition 3.2.2 provides the uniform convergence rate of the updated estimator $\hat{\boldsymbol{\zeta}}$, for any given $\boldsymbol{\zeta}^{init}$ in a a_n-neighborhood of $\boldsymbol{\zeta}^*$, and the linear characterization in result (ii) depicts the four pieces of estimation errors of $\hat{\boldsymbol{\zeta}}$: (2a) the systematic error from spline approximation and integration approximation, $O_p\left(\left(J_1^{-d}+J_2^{-d}+m_{*n}^{-1}\right)J_1^{1/2}\right)$; (2b) the oracle estimation error, $n^{-1}\mathbf{V}_2^{-1}\sum_{i=1}^{n}\rho_i\mathbf{W}_i$, obtained by knowing the nonparametric part $\eta_0(\cdot)$; (2c) the error coming from the oracle estimation of $\eta_0(\cdot)$, $n^{-1}\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}\sum_{j=1}^{n}\rho_j$ $\mathbf{B}_1\left(U_j\right)$; and (2d) the error brought by $\boldsymbol{\zeta}^{init}$,
$\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j){\mathbf{W}_j}^T\right]\left(\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right)+o_p(a_n)$, which is inherited from $\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^{init})$.

As the first three pieces of estimation errors, (2a) – (2c), are constant over iterations, a natural question is whether the last piece, (2d), would get amplified after iterations, leading to inconsistent estimations. By [19], we can show that

$$\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j){\mathbf{W}_j}^T\right]\leq\mathbf{I}_{(p+q)J_2\times(p+q)J_2},$$

and hence $\|\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j){\mathbf{W}_j}^T\right]\left(\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right)\|\leq\left\|\boldsymbol{\zeta}^{init}-\boldsymbol{\zeta}^*\right\|$, which implies that the estimation error from the initial $\boldsymbol{\zeta}^{init}$ would rather shrink over

iterations. This ensures the convergence of our algorithm.

Following [13], we need the initial $\boldsymbol{\zeta}^{init}$ to be $O(J_2^{1/2}n^{-1/2})$ consistent and the systematic error to be well bounded as $o(n^{-1/2})$ to investigate the properties of the final estimator $\hat{\boldsymbol{\zeta}}_f$. By Proposition 3.2.2, the final estimator $\hat{\boldsymbol{\zeta}}_f$ must satisfy

$$\hat{\boldsymbol{\zeta}}_f - \boldsymbol{\zeta}^* = n^{-1}\mathbf{V}_2^{-1}\sum_{i=1}^{n}\rho_i\mathbf{W}_i - n^{-1}\mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}\sum_{j=1}^{n}\rho_j\mathbf{B}_1\left(U_j\right) + \mathbf{V}_2^{-1}E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j)\mathbf{W}_j{}^T\right]\left(\hat{\boldsymbol{\zeta}}_f - \boldsymbol{\zeta}^*\right),$$

which yields a solution

$$\hat{\boldsymbol{\zeta}}_f = \boldsymbol{\zeta}^* + n^{-1}\Big(\mathbf{V}_2 - E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_i\mathbf{B}_1(U_i)\mathbf{W}_i{}^T\right]\Big)^{-1} \times\sum_{i=1}^{n}\rho_i\Big(\mathbf{W}_i - E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}\mathbf{B}_1\left(U_i\right)\Big). \tag{3.12}$$

The following theorem shows that the $\hat{\boldsymbol{\zeta}}_f$ obtained above is indeed the maximizer of the likelihood.

Theorem 3.2.1. *Under Conditions, let $\hat{\boldsymbol{\zeta}}_{mle}$ denote the maximizer of the likelihood function, then*

$$\hat{\boldsymbol{\zeta}}_{mle} - \boldsymbol{\zeta}^* = n^{-1}\mathbf{V}_3^{-1}\sum_{i=1}^{n}\rho_i\Big(\mathbf{W}_i - E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}\mathbf{B}_1\left(U_i\right)\Big).$$

and

$$\sqrt{n}\left(\hat{\boldsymbol{\zeta}}_{mle} - \boldsymbol{\zeta}^*\right) \xrightarrow{L} N\left(0, \mathbf{V}_3\right),$$

where $\mathbf{V}_3 = \mathbf{V}_2 - E\left[F_i\mathbf{W}_i\mathbf{B}_1^T(U_i)\right]\mathbf{V}_1^{-1}E\left[F_j\mathbf{B}_1(U_j)\mathbf{W}_j{}^T\right]$.

3.2.3 Proofs

Here we present the proofs of Theorems.

Conditions

Define $Q(x,y) = h(x)y - b(h(x))$, where h is the injective funciton, and denote $Q^{(l)} = (\partial^l/\partial x^l)Q(x,y), l = 1,2,3$. Then

$$Q^{(1)}(x,y) = h'(x)\left(y - \mu(h(x))\right)$$
$$Q^{(2)}(x,y) = h''(x)\left(y - \mu(h(x))\right) - \left(h'(x)\right)^2 \sigma(h(x))$$

Conditions:

A1 The function $Q^{(2)}(x,y) < 0$ for all $x \in \mathbb{R}$ and y in the range of the response variable.

A2 There exist two positive constants C_1 and C_2 such that $C_1 J_2^{-1} \leq \lambda_{\min}\left(E\left[(\boldsymbol{\phi}_i^T, \boldsymbol{\psi}_i^T)^T(\boldsymbol{\phi}_i^T, \boldsymbol{\psi}_i^T)\right]\right) \leq \lambda_{\max}\left(E\left[(\boldsymbol{\phi}_i^T, \boldsymbol{\psi}_i^T)^T(\boldsymbol{\phi}_i^T, \boldsymbol{\psi}_i^T)\right]\right) \leq C_2 J_2^{-1}$. This condition is analogous to the eigenvalue conditions widely adopted in multivariate literature. The order J_2^{-1} follows from Equation (12) in [19].

Proofs

We introduce some notations and state several results which will often be used in the proofs of our theoretical results.

According to [17], $\mathbf{B}_1^T = {\mathbf{B}_1^{\kappa-1}}^T \mathbf{D}$, where $\mathbf{B}_1^{\kappa-1} = (B_{s,1}^{\kappa-1}, 2 \leq s \leq J_1)^T$ is the B-spline basis functions with order $\kappa - 1$, and

$$\mathbf{D} = (\kappa-1)\begin{pmatrix} \frac{-1}{\tau_{\kappa+1}-\tau_2} & \frac{1}{\tau_{\kappa+1}-\tau_2} & 0 & \cdots & 0 & 0 \\ 0 & \frac{-1}{\tau_{\kappa+2}-\tau_3} & \frac{1}{\tau_{\kappa+2}-\tau_3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{-1}{\tau_{\kappa+J_1-1}-\tau_{J_1}} & \frac{1}{\tau_{\kappa+J_1-1}-\tau_{J_1}} \end{pmatrix}$$

is a $(J_1-1)\times J_1$ matrix. For $i=1,\cdots,n$, define

$$\gamma_i(\boldsymbol{\lambda},\boldsymbol{\zeta})=\boldsymbol{\lambda}^T\mathbf{B}_1\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}\right)+\boldsymbol{\psi}_i^T\boldsymbol{\omega},\quad \mathbf{W}_i(\boldsymbol{\lambda},\boldsymbol{\delta}):=\left(\boldsymbol{\lambda}^T\mathbf{B}_1'\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}\right)\boldsymbol{\phi}_i^T,\boldsymbol{\psi}_i^T\right)^T,$$

$$\theta_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=h\left(\gamma_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\right),\quad \mu_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=\mu(\theta_i(\boldsymbol{\lambda},\boldsymbol{\zeta})),\quad \sigma_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=\sigma(\theta_i(\boldsymbol{\lambda},\boldsymbol{\zeta})),$$

$$s_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):h'(\gamma_i(\boldsymbol{\lambda},\boldsymbol{\zeta}))\Big(Y_i-\mu_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\Big),\quad F_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=\sigma_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\left(h'\big(\gamma_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\big)\right)^2,$$

$$R_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=h''\big(\gamma_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\big)\Big(Y_i-\mu_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\Big),\quad H_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=F_i(\boldsymbol{\lambda},\boldsymbol{\zeta})-\mathbf{R}_i(\boldsymbol{\lambda},\boldsymbol{\zeta}),$$

$$\Psi_i(\boldsymbol{\lambda},\boldsymbol{\zeta}):=\frac{\partial\left(s_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\mathbf{W}_i(\boldsymbol{\lambda},\boldsymbol{\delta})\right)}{\partial\boldsymbol{\lambda}}$$

$$=-H_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\mathbf{W}_i(\boldsymbol{\lambda},\boldsymbol{\delta})\mathbf{B}_1^T(\boldsymbol{\phi}_i^T\boldsymbol{\delta})+s_i(\boldsymbol{\lambda},\boldsymbol{\zeta})\begin{pmatrix}\boldsymbol{\phi}_i\\0\end{pmatrix}\mathbf{B}_1'^T(\boldsymbol{\phi}_i^T\boldsymbol{\delta}).$$

In particular, we may write $\Lambda_i(\boldsymbol{\lambda}^*,\boldsymbol{\zeta}^*)$ as Λ_i^* for any $\Lambda_i(\boldsymbol{\lambda},\boldsymbol{\zeta})$ defined above.

By Conditions (C1), (C4), (3.4), and (3.6), it is easy to observe that

$$\begin{aligned}\left|\boldsymbol{\psi}_i^T\boldsymbol{\omega}^*-\int_{\mathcal{T}}\boldsymbol{\beta}_0^T(t)\mathbf{Z}_i(t)dt\right|&\\ &\le\left|\boldsymbol{\psi}_i^T\boldsymbol{\omega}^*-\sum_{l=1}^{m_i}(t_{i,l+1}-t_{i,l})\boldsymbol{\beta}_0^T(t_{i,l})\mathbf{Z}_i(t_{i,l})\right|\\ &\quad+\left|\sum_{l=1}^{m_i}(t_{i,l+1}-t_{i,l})\boldsymbol{\beta}_0^T(t_{i,l})\mathbf{Z}_i(t_{i,l})-\int_{\mathcal{T}}\boldsymbol{\beta}_0^T(t)\mathbf{Z}_i(t)dt\right|\\ &\le O\left(qJ_2^{-d}+m_{*n}^{-1}\right),\end{aligned}$$

$$\begin{aligned}\left|\boldsymbol{\lambda}^{*T}\mathbf{B}_1\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)-\eta_0\left(\int_{\mathcal{T}}\boldsymbol{\alpha}_0^T(t)\mathbf{X}_i(t)dt\right)\right|&\\ &\le\left|\boldsymbol{\lambda}^{*T}\mathbf{B}_1\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)-\eta_0\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)\right|\\ &\quad+\left|\eta_0\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)-\eta_0\left(\int_{\mathcal{T}}\boldsymbol{\alpha}_0^T(t)\mathbf{X}_i(t)dt\right)\right|\\ &\le O\left(J_1^{-d}+pJ_2^{-d}+m_{*n}^{-1}\right).\end{aligned}$$

By the mean value theorem, there exists a $\tilde{\mathbf{U}}_i$ between $\mathbf{U}_i$ and $\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*$, such that $\mathbf{B}_1(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*)-\mathbf{B}_1(U_i)=\mathbf{D}^T\mathbf{B}_1^{\kappa-1}(\tilde{U}_i)(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*-U_i)$. As $\|\mathbf{D}\|_1=O(J_1)$ and $\|\mathbf{B}_1^{\kappa-1}(\cdot)\|_\infty=$

$O(1)$, by (3.6) and (3.13), we have

$$\mathbf{B}_1(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*) = \mathbf{B}_1(U_i)(1 + O(J_1(J_1^{-d} + J_2^{-d} + m_{*n}^{-1})). \tag{3.13}$$

According to Equation (12) in [19],

$$C_1 J_1^{-1} \leq \lambda_{\min}\Big(E\left[\mathbf{B}_1\left(U_i\right)\mathbf{B}_1^T\left(U_i\right)\right]\Big) \leq \lambda_{\max}\Big(E\left[\mathbf{B}_1\left(U_i\right)\mathbf{B}_1^T\left(U_i\right)\right]\Big) \leq C_2 J_1^{-1}, \tag{3.14}$$

and consequently,

$$\begin{aligned} & C_1 J_1^{-1} \\ & \qquad \leq \lambda_{\min}\Big(E\left[\mathbf{B}_1\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)\mathbf{B}_1^T\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)\right]\Big) \\ & \qquad \leq \lambda_{\max}\Big(E\left[\mathbf{B}_1\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)\mathbf{B}_1^T\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)\right]\Big) \leq C_2 J_1^{-1}, \end{aligned} \tag{3.15}$$

By the definition of the l_r-norm of a matrix, (3.14), Condition (A2), and Condition (C3),

$$\left\|E[\mathbf{W}_i^*\mathbf{B}_1^T\left(U_i\right)]\right\| \vee \left\|E[\mathbf{W}_i^*\mathbf{B}_1^T\left(\boldsymbol{\phi}_i^T\boldsymbol{\delta}^*\right)]\right\| \leq C_2 J_1^{1/2} J_2^{1/2}. \tag{3.16}$$

Proof of Theorem 3.2.1: By Proposition 3.2.1, given any $\boldsymbol{\zeta}$ such that $\|\boldsymbol{\zeta} - \boldsymbol{\zeta}^*\| \leq a_n$, the log-likelihood function $L_n(\boldsymbol{\lambda}, \boldsymbol{\zeta})$ in (3.9) is maximized at

$$\begin{aligned} \hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}) = \boldsymbol{\lambda}^* + n^{-1}\mathbf{V}_1^{-1}\sum_{i=1}^{n}\rho_i\mathbf{B}_1\left(U_i\right) & \\ - \mathbf{V}_1^{-1}E\left[F_i\mathbf{B}_1(U_i)\mathbf{W}_i{}^T\right](\boldsymbol{\zeta} - \boldsymbol{\zeta}^*) & \\ + o_p(b_n J_1^{1/2}). & \end{aligned} \tag{3.17}$$

Thus, $L_n(\boldsymbol{\lambda}, \boldsymbol{\zeta})$ can be considered as a function of only one argument $\boldsymbol{\zeta}$ only, namely

$$L_n(\boldsymbol{\lambda}, \boldsymbol{\zeta}) = \ell_n(\boldsymbol{\zeta}) = \sum_{i=1}^{n} h\big(\gamma_i(\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}), \boldsymbol{\zeta})\big)Y_i - b\left(h\left(\gamma_i(\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}), \boldsymbol{\zeta})\right)\right). \tag{3.18}$$

By a Taylor expansion of $\ell_n(\boldsymbol{\zeta})$, we obtain that

$$\ell_n(\boldsymbol{\zeta}) - \ell_n(\boldsymbol{\zeta}^*) = \left.\frac{\partial \ell_n(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right|_{\boldsymbol{\zeta}^*}(\boldsymbol{\zeta} - \boldsymbol{\zeta}^*) + \frac{1}{2}(\boldsymbol{\zeta} - \boldsymbol{\zeta}^*)^T\left.\frac{\partial^2 \ell_n(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}\partial \boldsymbol{\zeta}^T}\right|_{\breve{\boldsymbol{\zeta}}}(\boldsymbol{\zeta} - \boldsymbol{\zeta}^*),$$

where $\breve{\boldsymbol{\zeta}}$ is a point between $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}^*$.

By (3.17), we obtain that

$$\frac{\partial \hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = \mathbf{V}_1^{-1} E\left[F_i \mathbf{B}_1(U_i) \mathbf{W}_i{}^T\right](1 + o_p(1)). \tag{3.19}$$

and consequently

$$\begin{aligned}\left.\frac{\partial \ell_n(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right|_{\boldsymbol{\zeta}^*} &= \sum_{i=1}^{n} h'\Big(\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^*)^T \mathbf{B}_1\left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}^*\right) + \boldsymbol{\psi}_i^T \boldsymbol{\omega}^*\Big) \\ &\quad \Big(Y_i - \mu\big(h(\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^*)^T \mathbf{B}_1\left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}^*\right) + \boldsymbol{\psi}_i^T \boldsymbol{\omega}^*)\big)\Big) \\ &\quad \times \left\{\left(\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^*)^T \mathbf{B}_1'\left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}^*\right) \boldsymbol{\phi}_i^T, \boldsymbol{\psi}_i^T\right)^T - \left(\mathbf{B}_1^T\left(\boldsymbol{\phi}_i^T \boldsymbol{\delta}^*\right) \mathbf{V}_1^{-1} E\left[F_i \mathbf{B}_1(U_i) \mathbf{W}_i{}^T\right]\right)^T\right\}.\end{aligned}$$

By Proposition 3.2.1, $\hat{\boldsymbol{\lambda}}(\boldsymbol{\zeta}^*) - \boldsymbol{\lambda}^* = n^{-1}\mathbf{V}_1^{-1}\sum_{i=1}^{n} \rho_i \mathbf{B}_1(U_i) = O_p(J_1^{1/2} n^{-1/2})$. Then by the arguments used in Propositions 3.2.1 and 3.2.2, we can show that

$$\left.\frac{\partial \ell_n(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right|_{\boldsymbol{\zeta}^*} = \sum_{i=1}^{n} \rho_i \Big(\mathbf{W}_i - \Big(E\left[F_i \mathbf{W}_i \mathbf{B}_1^T(U_i)\right]\Big)\mathbf{V}_1^{-1}\mathbf{B}_1(U_i)\Big)(1 + o_p(1)).$$

Similarly, it can be shown that

$$\left.\frac{\partial^2 \ell_n(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta} \partial \boldsymbol{\zeta}^T}\right|_{\breve{\boldsymbol{\zeta}}} = -n\mathbf{V}_3(1 + o_p(1)).$$

Therefore, we obtain that

$$\hat{\boldsymbol{\zeta}}_{mle} = n^{-1}\mathbf{V}_3^{-1}\sum_{i=1}^{n} \rho_i \Big(\mathbf{W}_i - \Big(E\left[F_i \mathbf{W}_i \mathbf{B}_1^T(U_i)\right]\Big)\mathbf{V}_1^{-1}\mathbf{B}_1(U_i)\Big).$$

Then according to Central Limit Theorem,

$$\sqrt{n}\left(\hat{\boldsymbol{\zeta}}_{mle} - \boldsymbol{\zeta}^*\right) \xrightarrow{L} N(0, \mathbf{V}_3^{-1}),$$

This completes our proof of Theorem 3.2.1. □

Lemma 3.2.1. *Under Condition (C1), there exists* $\tilde{\boldsymbol{\delta}}_1$ *with* $\tilde{\boldsymbol{\delta}}_{1,1} \le \tilde{\delta}_{2,1} \le \cdots \le \tilde{\delta}_{J_2,1}$ *such that* $\|\alpha_1 - \tilde{\boldsymbol{\delta}}_1^T \mathbf{B}_2(t)\|_\infty = O(J_2^{-d})$.

Proof: Let $\{\epsilon_r, r = 1, \cdots, J_2\}$ be a sequence on $\mathcal{T}$, such that $\epsilon_r = \nu_1 + (r -$

$1)(\nu_{\kappa+1}-\nu_\kappa)/\kappa$ for $r=1,\cdots,\kappa$ and $\epsilon_r=\nu_r$ for $r=\kappa+1,\cdots,J_2$. Since $\alpha_1(t)$ is monotone nondecreasing in t, $\alpha_1(\epsilon_r)$ is monotone nondecreasing in r. Define $\tilde{\alpha}_{1,n}(t)=\sum_{r=1}^{J_2}\alpha_1(\epsilon_r)B_{r,2}(t)$. Given any $t\in[\nu_l,\nu_{l+1})$,

$$|\tilde{\alpha}_{1,n}(t)-\alpha_1(t)|\le\sum_{r=1}^{J_2}|\alpha_1(\epsilon_r)-\alpha_1(t)|\,B_{r,2}(t)=\sum_{r=l+1-\kappa}^{l}|\alpha_1(\epsilon_r)-\alpha_1(t)|\,B_{r,2}(t)$$

$$\le\max_{l+1-\kappa\le r\le l}|\alpha_1(\epsilon_r)-\alpha_1(t)|\le C_D\max_{l+1-\kappa\le r\le l}|\epsilon_r-t|^d\le C_D(\kappa+1)^d\left(\max_{\kappa\le r\le J_2}|\nu_{r+1}-\nu_r|\right)^d$$

$$=O(J_2^{-d}),$$

where the two inequalities are elementary, the first equality follows from the fact that $B_{r,2}(t)=0$ for all $r<l+1-\kappa$ and $r>l$, the third inequality follows from Condition (C1), and the fourth inequality follows from the fact $|\epsilon_r-t|\le(\kappa+1)\max_{\kappa\le r\le J_2}|\nu_{r+1}-\nu_r|$, and the second equality follows from our choice of $\{\nu_r\}$. This completes the proof of Lemma 3.2.1. □

3.2.4 Simulations

In these simulations we examine the finite sample performance of the generalized functional partially linear single index model (GFPL). We explore both logistic and poisson regression models. The logistic model includes a binary outcome variable and the poisson model has a count outcome variable. Both set ups have two functional predictor variables. The predictor variables are defined as,

$$X_{i,k}(t)=t+\sum_{j=1}^{4}\phi_{j,k}(t)'c_{ijk}, \tag{3.20}$$

where k=1, 2; i=1,...,n; and c_{ijk} are i.i.d $N(0,\sigma_j^2)$. We define $\sigma_1^2=1,\sigma_2^2=\frac{1}{2},\sigma_3^2=\frac{1}{4},\sigma_4^2=\frac{1}{8}$ and $\phi_{1,k}(t)=\frac{1}{\sqrt{2}}\sin(2\pi t),\phi_{2,k}(t)=\frac{1}{\sqrt{2}}\cos(2\pi t),\phi_{3,k}(t)=\frac{1}{\sqrt{2}}\sin(4\pi t),\phi_{4,k}(t)=\frac{1}{\sqrt{2}}\cos(4\pi t)$. We let n = (200, 400, 800) (the number of observations), and m=100 evenly spaced time points in [0, 1]. We also define, $\beta_1(t)=\cos(\pi t+\pi)$ and $\beta_2(t)=\sin(\pi t)$ such that $||\beta_1||+||\beta_2||=1$. Next, we define $U_i=\frac{1}{m}[\sum_{k=1}^{2}\beta_k(t)X_{i,k}(t)dt]$.

We then generate the responses from the GFPL as $\ln(\frac{p_i}{1-p_i}) = \eta_0(U_i)$, for the logistic setting and $ln(y_i) = \eta_0(U_i)$ for the poisson setting using four different link functions, η_0, defined as follows and plotted in Figure 3.1:

1. $5U_i + 1$
2. $\sin(\pi(U_i - U_{min})/(U_{max} - U_{min}) - \frac{\pi}{2})$
3. $\frac{U_i^2}{5} + \frac{exp(U_i)}{5} - 2\sin(2\pi U_i) + 0.5$
4. $\frac{U_i^2}{5} + 2\cos(2\pi U_i) + 0.5$

Using the generated probabilities, we generate our true responses from a $binomial(n, 1, p_i)$ distribution for the logistic setting; from a $poisson(n, p_i)$ distribution for the poisson setting. We used cubic B-splines with order q=4 and N= 3 knots to approximate the nonparametric functions $\eta_0(\cdot)$ and $\beta_k(\cdot)$.

We ran 100 simulations, splitting the data into training and test sets using a 50/50 split. For each set up we calculated the mean standard error for $\hat{\beta}_1 = \frac{\sum_{t=1}^{m}[\beta_1(t)-\hat{\beta}_1(t)]^2}{m}$ and $\hat{\beta}_2 = \frac{\sum_{t=1}^{m}[\beta_2(t)-\hat{\beta}_2(t)]^2}{m}$; for the logistic setting we also calculated the mean correct predicted classification percentage and compare our results to the functional generalized linear model using functional principal component scores described in Project 1. The results are summarized in Tables 3.1-3.6.

3.2.5 Application

Lastly, we apply our methods to the Lupus dataset described in Project 1. In this application, we compare our proposed method to the functional generalized linear model using functional principal component scores. To approximate the data, we use q=4 order B-splines for our method and the first four principal components for the FGLM with FPCA model. We then split the data into training and test sets using a $\frac{2}{3}$ vs. $\frac{1}{3}$ split; yielding 200 cases and 200 controls in the training set and 98 cases and

94 controls in the test set. We repeat this splitting 100 times and calculate the mean correct predicted classification percentage. These results are summarized in Table 3.7.

3.3 Results

3.3.1 Simulations

From Tables 3.1-3.3, in the Logistic setting, we see that our methods perform slightly worse than the FGLM with FPCA for the first two link functions. The first link function is strictly linear, so we would expect the standard FGLM to perform best in this setting. The second link function is almost linear so it is not a surprise that the FGLM is able to still perform well in this setting. The last two link functions are much more non-linear than the first two. Here we see that our model performs significantly better than the FGLM. The GFPL has a significantly higher prediction accuracy and much more stable regression coefficient estimates than the FGLM.

From Tables 3.4-3.6, in the Poisson setting, we see a similar trend as we saw in the Logistic setting. The two models yield comparable results for the first two link functions. Here, the GFPL yields a more stable β_1 estimate, whereas the FGLM yields a more stable β_2 estimate. However, when looking at the last two link functions we see the same trend we saw with the Logistic set up - the GFPL produces much more stable estimates of the regression coefficients than the FGLM.

In conclusion, as greater non-linearity is introduced, the GFPL is able to estimate the correct injective link function between the logit and the predictor variables. The FGLM is not able to capture this non-linear relationship. Therefore, the non-linearity is captured in the regression coefficient estimates leading to much larger, more unstable estimates than we see in the GFPL. Additionally, this leads to reduced prediction accuracies for the FGLM in the logistic setting. We can conclude that if the

assumption of a linear relationship between the logit and the predictors is violated, the GFPL would be best.

3.3.2 Application

From Table 3.7 we see that our model has a slightly higher correct classification rate compared to the FGLM with FPCA model. While this difference is not significant we can conclude that our methods perform as well as the standard FGLM with FPCA in this setting. We also note that the correct classification rate is not as high as we would hope. We believe this may be due to a high level of noise from including all functional points on the curve and that only some parts of the curve are truly important. This leads us to the motivation for Project 3 which is to develop a model that is able to determine where the true signal lies.

3.4 Tables and Figures

Table 3.1: Simulation Results for logistic setting with n=200

η_0	GFPL			FGLM		
	Pred	SE(β_1)	SE(β_2)	Pred	SE(β_1)	SE(β_2)
$5U + 1$	80.42%	0.11 (0.08)	0.15 (0.15)	82.27%	0.29 (0.13)	0.12 (0.13)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))} - \frac{\pi}{2}\right)$	55.09%	0.14 (0.09)	0.25 (0.22)	56.87%	0.41 (0.06)	0.10 (0.05)
$\frac{U^2}{5} \times \frac{\exp(U)}{5} - 2\sin(2\pi U) + 0.5$	61.61%	0.10 (0.09)	0.12 (0.20)	53.35%	0.56 (0.10)	0.13 (0.07)
$\frac{U^2}{5} + 2\cos(2\pi U) + 0.5$	65.39%	0.09 (0.10)	0.12 (0.17)	58.86%	0.66 (0.11)	0.17 (0.06)

Table 3.2: Simulation Results for logisitc setting with n=400

η_0	GFPL			FGLM		
	Pred	SE(β_1)	SE(β_2)	Pred	SE(β_1)	SE(β_2)
$5U+1$	81.57%	0.08 (0.08)	0.12 (0.11)	82.60%	0.23 (0.05)	0.05 (0.04)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))}-\frac{\pi}{2}\right)$	54.87%	0.13 (0.10)	0.22 (0.20)	56.72%	0.41 (0.05)	0.09 (0.03)
$\frac{U^2}{5}\times\frac{\exp(U)}{5}-2\sin(2\pi U)+0.5$	65.78%	0.06 (0.07)	0.06 (0.11)	54.90%	0.52 (0.05)	0.12 (0.04)
$\frac{U^2}{5}+2\cos(2\pi U)+0.5$	65.68%	0.08 (0.11)	0.12 (0.23)	60.77%	0.63 (0.07)	0.14 (0.04)

Table 3.3: Simulation Results for logisitc setting with n=800

η_0	GFPL			FGLM		
	Pred	SE(β_1)	SE(β_2)	Pred	SE(β_1)	SE(β_2)
$5U+1$	82.59%	0.05 (0.05)	0.07 (0.06)	83.46%	0.23 (0.03)	0.03 (0.02)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))}-\frac{\pi}{2}\right)$	56.26%	0.10 (0.08)	0.22 (0.16)	57.61%	0.41 (0.03)	0.08 (0.02)
$\frac{U^2}{5}\times\frac{\exp(U)}{5}-2\sin(2\pi U)+0.5$	67.87%	0.06 (0.08)	0.04 (0.13)	56.41%	0.53 (0.03)	0.11 (0.02)
$\frac{U^2}{5}+2\cos(2\pi U)+0.5$	70.35%	0.07 (0.05)	0.05 (0.13)	61.53%	0.61 (0.04)	0.13 (0.03)

Table 3.4: Simulation Results for poisson setting with n=200

η_0	GFPL		FGLM	
	SE(β_1)	SE(β_2)	SE(β_1)	SE(β_2)
$5U + 1$	0.08 (0.07)	0.12 (0.12)	0.22 (0.01)	0.02 (0.01)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))} - \frac{\pi}{2}\right)$	0.11 (0.08)	0.18 (0.16)	0.41 (0.03)	0.08 (0.02)
$\frac{U^2}{5} \times \frac{\exp(U)}{5} - 2\sin(2\pi U) + 0.5$	0.05 (0.08)	0.08 (0.15)	0.49 (0.05)	0.11 (0.03)
$\frac{U^2}{5} + 2\cos(2\pi U) + 0.5$	0.04 (0.05)	0.04 (0.17)	0.60 (0.04)	0.13 (0.03)

Table 3.5: Simulation Results for poisson setting with n=400

η_0	GFPL		FGLM	
	SE(β_1)	SE(β_2)	SE(β_1)	SE(β_2)
$5U + 1$	0.08 (0.05)	0.08 (0.09)	0.22 (0.01)	0.02 (0.01)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))} - \frac{\pi}{2}\right)$	0.08 (0.08)	0.15 (0.12)	0.41 (0.02)	0.07 (0.01)
$\frac{U^2}{5} \times \frac{\exp(U)}{5} - 2\sin(2\pi U) + 0.5$	0.04 (0.06)	0.03 (0.12)	0.49 (0.03)	0.10 (0.02)
$\frac{U^2}{5} + 2\cos(2\pi U) + 0.5$	0.06 (0.07)	0.04 (0.12)	0.58 (0.03)	0.12 (0.02)

Table 3.6: Simulation Results for poisson setting with n=800

η_0	GFPL		FGLM	
	SE(β_1)	SE(β_2)	SE(β_1)	SE(β_2)
$5U + 1$	0.05 (0.04)	0.07 (0.06)	0.21 (0.01)	0.02 (0.01)
$\sin\left(\frac{\pi(U-\min(U))}{(\max(U)-\min(U))} - \frac{\pi}{2}\right)$	0.07 (0.06)	0.11 (0.08)	0.41 (0.01)	0.07 (0.01)
$\frac{U^2}{5} \times \frac{\exp(U)}{5} - 2\sin(2\pi U) + 0.5$	0.03 (0.04)	0.02 (0.02)	0.49 (0.03)	0.10 (0.01)
$\frac{U^2}{5} + 2\cos(2\pi U) + 0.5$	0.05 (0.07)	0.02 (0.12)	0.58 (0.02)	0.12 (0.01)

Table 3.7: Prediction accuracies of each method

Method	**Accuracy**
FGLM	70.73% (0.05)
GFPL	71.49% (0.02)

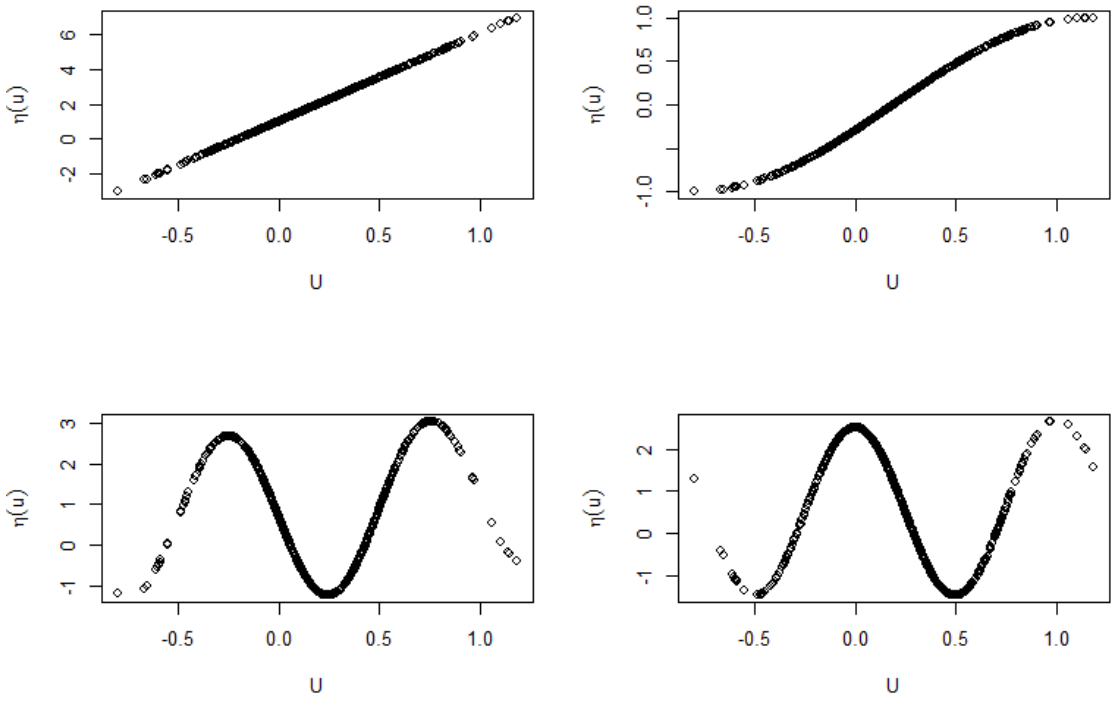

Figure 3.1: The four link functions for simulations

CHAPTER 4

LOCAL BASIS RANDOM FORESTS FOR FUNCTIONAL DATA

4.1 Introduction

The main characteristic of functional data is the intercorrelation between consecutive features. As a result, direct use of traditional multivariate machine learning classifiers face the difficulty of multicollinearity, and therefore may not have satisfactory results. For example, most linear models in this case will suffer high variances on their parameter estimates. Naturally, principal component analysis (PCA) arises for dealing with correlated features. As a data preprocessing tool, PCA considers correlated features collectively and finds the directions with the largest variations. By projecting the original features onto these PC directions, performance of the traditional classifiers can sometimes be improved. However, in many classification problems, not all features are relevant to predicting the class label. For functional data, signals can concentrate on some local regions of the entire curve only. Since, in PCA, each principal component is a linear combination of all features, it sometimes mixes those relevant features and makes it difficult to assess the importance of the original features. For example, as we saw in Project 1 and as we see in Figure (4.1(a)), only a local region (60 - 70°C) received a high importance estimate. On the other hand, features at other temperatures seem to provide little discriminatory information. In this situation variable selection is needed and using principal components, as we did in Projects 1 and 2, that involve all variables may fail to emphasize on those important features.

Decision tree based methods have been shown as powerful tools to conduct variable selection and classify groups [21, 22, 23]. A traditional decision tree is constructed by recursively splitting the data set. At each node, one single variable is selected for splitting the data within that node. Usually, the splitting variable is the one that carries the most discriminatory information about the two classes. Since, at each time, only one variable is considered, the variance of the effect of that variable is not inflated by multicollinearity. Meanwhile, tree based methods enjoy several other advantages including, but not limited to: 1. Nonparametric; 2. relatively fast computation time; 3. stability can be increased by using an ensemble; 4. able to deal with nonlinear decision boundaries.

We propose to develop a decision tree method that can utilize the discriminatory information in $\boldsymbol{X}_i$ to predict the class of Y_i. There have been numerous related works in the literature: [24] proposed an approach to make inference about the probability density of functional data through what they call density ascent lines; [25] employed a two-stage shrinkage method in the functional linear regression model to handle the situation where certain regions of the functional curve have a zero coefficient function; and [26] proposed a kernel-induced random forest approach for the functional data classification problem.

We want to examine a nonparametric approach and use local information for prediction. Therefore we consider a tree-based method. However, the traditional decision tree method has disadvantages when it comes to functional data. Because of the univariate type of splits, the resulting decision boundary must be perpendicular to certain variable axes. Since, for functional data, due to the intercorrelation among functional features, data points lie mostly along the directions of diagonal lines in the hyperspace, the hyper-rectangular decision boundaries may not be the most efficient ones. There are several ways to remedy this situation. First, [27] uses a penalized linear model within each tree node to find a linear combination of features. The set of

linear coefficients determines a splitting direction that is adapted to the locally best class separation. In our study, we find their method is generally competitive at classifying functional data; however, using linear models to determine splitting direction puts a limit on its application to nonlinear decision boundaries. In our simulation study, this method fails in some nonlinear cases. Another remedy is the rotation forest [28]. Rodriquez, et al. proposed applying block-wise PCA to the original feature matrix. Therefore, the resulting splitting direction is a linear combination of original features that is adapted to data variation. However, in their method, all features are randomly divided into several groups, and PCA is applied within each group. Thus, this method does not account for the continuity of the functional data. We propose a PCA-based local basis expansion method

4.2 Method

4.2.1 Proposed Method

Notation and tree-based method

Assume that for each observation we have a binary response variable Y_i, where i denotes the index of the observation, and a px1 feature vector $\boldsymbol{X}_{i\cdot} = X_{ij}$, where $1 \leq j \leq p$, is the jth feature of the ith observation.

In the decision tree algorithm, the feature space is partitioned into several non-overlapping bins. This is done by recursively splitting the training data based on a certain splitting rule. For example, in a traditional decision tree, at a node $\mathcal{A}$, the splitting rule is that we select a certain variable, say X_k, as our splitting variable, where $1 \leq k \leq p$, and the training data within node $\mathcal{A}$ are divided into two parts according to the value of X_k. That is, we set a critical value c, and if $X_k > c$ is satisfied for an observation, this observations goes to the left child of $\mathcal{A}$, otherwise it goes to $\mathcal{A}$'s right child. Then, for the left and right child of $\mathcal{A}$, we repeat

this procedure until all of the leaf nodes are small enough, in which case we call it a terminal node. In practice the splitting variable at each node is selected from a subset of all features, and the critical value, c, is searched over all possible splits of X_k. Usually, the combination of X_k and c that brings the largest decrease in the Gini impurity after the split will be employed as the splitting rule. Furthermore, if we construct a large number of trees, each of which is fitted using a bootstrap sample of the training data, we get the so-called random forest algorithm.

For predicting the response of a new observation, we drop this new observation down the tree and see which bin it falls into. The same splitting rule is used here to determine which child node the new observation goes to. In other words, if at node $\mathcal{A}$, $X_k > c$ is used for splitting the training data, then the new observation goes to the left child of $\mathcal{A}$ if $X_k > c$ is also true for the new observation, otherwise it goes to $\mathcal{A}$'s right child. This way the new observation keeps going down the tree until reaching a terminal node. Then the averaged value of the responses of the training data within that terminal node will be output as the predicted value. For the random forest algorithm, the output of each tree is again averaged. In the context of classification, taking the average is replaced by majority vote.

To facilitate our discussion, we first need to formalize our notation for a general splitting rule. Let $\boldsymbol{b}$ denote a px1 vector, and let c denote the critical value. Then, for the ith observation, a general splitting rule of a decision tree can be written as

$$\boldsymbol{X}_i(t)\boldsymbol{b} > c \tag{4.21}$$

In other words, $\boldsymbol{b}$ is a coefficient vector that determines the splitting direction in the hyperspace.

Equation 4.21 is a general form of the splitting rule of a tree, and it can be specialized to various kinds of splitting rules. For example, when only one element of $\boldsymbol{b}$ is non-zero while all other elements are 0, Equation 4.21 boils down to the univariate

splitting rule as in traditional random forest. In the case of oblique forest, $\boldsymbol{b}$ is the coefficients from either a linear model or support vector machine (SVM), and in the rotation forest, $\boldsymbol{b}$ is the loading vector produced by the block-wise PCA. For our task, simply speaking, we need to find a set of coefficients that works the best for functional covariates.

Proposed functional splitting rule

One suitable tool for processing functional data is principal component analysis (PCA). First, it decorrelates features. In the case of functional data, as the features next to each other tend to vary together, PCA finds the direction that simultaneously explains their joint variation. Also, the principal components can be viewed as a set of basis functions used to expand the curve. Compared to other basis expansion approaches, PCA is particularly effective when we have no idea what important features may look like since PCA identifies features that can explain data variability well and automatically [29]. Meanwhile, it can be shown that for a fixed number of basis functions, principal components have the smallest mean squared error among all basis function approximations [7]. In this paper, we propose a PCA-based local basis expansion method. Essentially, at each split we focus on a local region of the curve and apply PCA on the local curve. The loading vector of the principal components provides possible splitting directions. Equivalently, speaking in terms of Equation 4.21, we select two indices, s and e such that $1 \leq s \leq e \leq p$, and we set the constraint that $\boldsymbol{b}_j = 0$ for $1 \leq j < s$ and $e < j \leq p$, and $\boldsymbol{b}_s$ through $\boldsymbol{b}_e$ are determined by the loading vector obtained from the local PCA. The pseudocode in Algorithm 1 explains our proposed method.

There are several tuning parameters in our method. The *bandwidth* determines how many features are present in a band. In our study, *bandwidth* is generated randomly and uniformly from 1 to *bandwidth.max*, where *bandwidth.max* is a tuning

parameter. *mtry* is analogous to the *mtry* in the traditional random forest algorithm. It is the number of different bands that are sampled at each node. In our algorithm, we set $mtry = 1.5 \times p/bandwidth$. The intuition is that the smaller the *bandwidth*, the more bands we need to sample in order to have a good cover of the whole curve. L is the number of principal components to keep. By keeping the first few principal components, we hope to filter out the noises.

One issue regarding PCA is that as the sample size becomes small, the directions learned from PCA will become dominated by noise, and will lose the ability to capture the true signals. In our tree algorithm, at each node, we only use the observations within the current node to calculate the sample covariance matrix, and PCA is carried out on this covariance matrix. As we go down the tree, the sample size becomes smaller and smaller. Near terminal nodes the directions obtained from PCA are too noisy. In order to deal with this situation, we set a threshold parameter, m. Suppose at node $\mathcal{A}$ the sample size is less than or equal to m, then for all of the descendent nodes of $\mathcal{A}$, PCA will be done on the current covariance matrix at $\mathcal{A}$. In the following analysis, we set $m = log(n)$, where n is the sample size of the training data.

The main advantages of LBRF are three-fold: 1. The PCA-based splitting rule handles the correlation among functional features; 2. The local basis expansion approach enables LBRF to focus on the local region where the true signal lies, and prevents the true signal from being mixed with misleading noises; 3. The multiple candidate splitting locations allows LBRF to adaptively choose the location with the strongest signals. In essence, LBRF is an analog to the traditional random forest algorithm; the only difference being that instead of sampling one variable, LBRF samples a band of features at a time.

Algorithm 1: splitting a node

Input: Tree node $\mathcal{A}$
if $n_{\mathcal{A}} \leq nodesize$ **then**
  Output: $\mathcal{A}$ is a terminal node
end
for *i from 1 to mtry* **do**
  Let $X_{\mathcal{A}}$ denote the feature matrix at node $\mathcal{A}$. Randomly sample a band of features of length *bandwidth*. Form a submatrix of $X_{\mathcal{A}}$, with the columns corresponding to the sampled features. Denote this submatrix by X_{sub}
  Apply PCA on X_{sub}. Keep the first L principal components, which we denote by $(PC_1^{(i)}, PC_2^{(i)}, \ldots, PC_L^{(i)})$
  for *j from 1 to L* **do**
    Randomly generate a splitting point of $PC_j^{(i)}$ that divides $PC_j^{(i)}$ into two parts, calculate the corresponding reduction in Gini impurity, and denote it by g_{ij}
  end
end
Let i_{max} and j_{max} be the indices of the largest g_{ij}. Then split corresponding to$PC_{jmax}^{(imax)}$ is used to split the data at $\mathcal{A}$. The resulting two parts of the data are passed down to the two children nodes of $\mathcal{A}$
Output: Return the two children nodes

Variable Importance

In many machine learning applications, variable selection is often as important as prediction accuracy. In practice, smaller models are usually preferred for their simplicity and interpretability. Many modeling approaches come with a variable selection mechanism. For example, in penalized linear models, when the Lasso penalty is used, some coefficients are shrunk to 0, meaning the corresponding variables don't play any role in predicting the outcome and can be dropped from the model.

The random forest algorithm is able to perform variable selection through its variable importance. In random forest, there are two types of variable importance. The first is permutation importance. This type of importance is calculated by permuting the values of the out-of-bag (OOB) samples for a given variable, dropping the original/permuted (OOB) data down the tree, and calculating the decrease in

prediction accuracy before and after the permutation. The second one is Gini importance, which is the decrease in Gini impurity after splitting a node. In terms of computational efficiency, Gini importance is easier to calculate since it doesn't involve permuting data and walking each OOB observation down the tree.

The oblique forest also implemented another kind of variable importance by calculating the analysis of variance table at each split and counting for each variable how often it is considered significant in the ANOVA table [26].

The rotation forest was constructed by projecting the original data to the principal component directions and fitting a decision tree to the resulting data. The Gini importance of each PC can be calculated on the transformed data. Then these importance measures can be mapped back to the original features through the loading vector of the PC.

We develop a variable importance measure that is similar to the Gini importance in the random forest. Algorithm 2 shows the procedure of calculating variable importance.

Algorithm 2: Variable Importance

Input: Tree $\mathcal{T}$
Initialize $\boldsymbol{VI}_{\mathcal{T}}$ to be a length p vector of 0s
for *Node* $\mathcal{I}$ *in* $\mathcal{T}$ **do**
  if *Node* $\mathcal{I}$ *has decendents* **then**
    Calculate the decrease in Gini impurity after the split $\mathcal{I}$, which we denote by $d_{\mathcal{I}}$
    Normalize the coefficient vector $\boldsymbol{b}_{\mathcal{I}}$, which was used for the splitting node $\mathcal{I}$ (Equation 4.21), to a unit vector
    Set $\boldsymbol{VI}_{\mathcal{T}} = \boldsymbol{VI}_{\mathcal{T}} + d_{\mathcal{I}}\boldsymbol{b}_{\mathcal{I}}$
  end
end
Normalize $\boldsymbol{VI}_{\mathcal{T}}$ to a unit vector
Output: Return $\boldsymbol{VI}_{\mathcal{T}}$ as the vector of variable importance of tree $\mathcal{T}$

For an ensemble of trees, we take the element-wise average of $VI_{\mathcal{T}}$ over all trees, and output the resulting length p vector as our variable importance measure.

Simulation study

Our first focus is the cross validation accuracy. We test the performance of our tree algorithm on four simulated functional datasets and we compare its cross validation accuracy with the penalized linear model (from "glmnet" R package), random forest (from "randomForest" R package), oblique forest (from "obliqueRF" R package), and rotation forest (from "rotationForest" R package). The four datasets represent four different data generating scenarios. Let t be the index variable of the simulated random functions. In our simuation, t ranges from 1 to 10, and is discretized into 400 equally spaced time points (thus, $p = 400$). Then we simulate the deterministic curve $X_0(t) = 5sin(t)$, and the noise term $Z(t) = \mu_1 sin(\alpha_1+\beta_1 t)+\mu_2 sin(\alpha_2+\beta_2 t)+\epsilon(t)$. The parameters $\mu_1, \mu_2, \alpha_1, \alpha_2, \beta_1, \beta_2$ are independent normal random variables randomly generated for each observation. The first two terms of $Z(t)$, which are two random sinusoidal waves, can be viewed as a functional noise component. They provide a unique shape change to each individual base curve, and the intercorrelations among functional covariates are also induced by these terms. $\epsilon(t)$ is a series of i.i.d normal random variables; they mimic the measurement errors. We let $F(t) = X_0(t) + Z(t)$. The four different scenarios are:

Scenario 1: For class 0,

$$X(t) = F(T); \tag{4.22}$$

For class 1,

$$X(t) = F(t) + \mathbb{1}_{5.25\leq t\leq 5.71} 0.85 e^{48.94\times(t-5.5)^2}, \tag{4.23}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. The difference between the two classes is that class 1 has a local bump at the region $5.28 \leq t \leq 5.71$

Scenario 2: For class 0,

$$X(t) = X_0(t) + Z(t). \tag{4.24}$$

For class 1,

$$X(t) = 1.06X_0(t) + Z(t). \tag{4.25}$$

Class 1 has a larger amplitude for the base curve than class 0.

Scenario 3: Both class 0 and class 1 have the curve

$$\begin{aligned} X(t) = \; & F(t) \\ & + \alpha_1 \mathbb{1}_{3.03 \le t \le 3.46}(e^{-23.782(t-3.24)^2} - 0.0781) \\ & + \alpha_2 \mathbb{1}_{5.28 \le t \le 5.71}(e^{-23.782(t-5.5)^2} - 0.0781) \\ & + \alpha_3 \mathbb{1}_{7.54 \le t \le 7.97}(e^{-23.782(t-7.76)^2} - 0.0781), \end{aligned} \tag{4.26}$$

where $\alpha_1, \alpha_2, \alpha_3 \sim U(0, 2)$. In other words, there are three bumps of width approximately 0.43 that are centered at $t = 3.24, t = 5.5, t = 7.76$. The class label is 1 if all α_1, α_2, and α_3 are greater than 1.587.

Scenario 4: Both class 0 and class 1 have the curve

$$X(t) = F(t). \tag{4.27}$$

There are four possible bumps that are centered at $t = 2.12, t = 4.37, t = 6.63$, and $t = 8.88$. A bump is realized by adding $0.9e^{-48.94\times(t-c)^2}$, where c is the center of the bump. Their occurrence probabilities are mutually independent. The class label of an observation is 1 if either the first bump or the last bump, or both of them occur; otherwise the class label is 0. The second and third bumps are merely noise.

Figure 4.2 shows each simulation scenario with the base curve and noises removed.

We run each classifier with several different sets of tuning parameters. Table 4.1 shows all combinations of tuning parameters we consider. Parameters that stay the same throughout the simulation are omitted from the table. For each simulation scenario, we generate 1200 samples, 200 of which are randomly selected for training the classifiers, and the remaining 1000 samples are used for testing the performance.

We repeat this training-testing splitting 50 times for a given set of tuning parameters, and the averaged prediction accuracy is used to evaluate the classifiers.

As we have noted, PCA is frequently used for data preprocessing. FPCA projects the functional data onto principal component directions and drops the PCs with small variance. This way, we hope to filter out noise. In our simulation, for the penalized linear model and random forest approaches, we also consider projecting the original features onto the PC directions and then feed the first 50 PC scores to the classifiers. Details are shown in Table 4.1.

4.3 Results

4.3.1 Simulation Results: Cross Validation Accuracy

Figure 4.3 uses a box plot to show the performance of different classifiers in each simulation scenario. Each box depicts all cross validation accuracies of the classifier under different sets of tuning parameters (listed in Table 4.1). In Table 4.2, for each classifier we pick the set of tuning parameters which yielded the highest accuracy among all combinations of tuning parameters from Table 4.1, and present the resulting mean predicition accuracy in the table. In the parentheses is the corresponding standard deviation estimated from the 50 runs.

We can see from Table 4.2 that for Scenario 1, 3, and 4, LBRF achieved the highest prediction accuracy, which demonstrates the ability of LBRF to utilize localized discriminatory signals. The improvement brought by LBRF over traditional random forest is at least two times their standard deviations, which is significant. Scenarios 3 and 4 both possess nonlinear decision boundaries. In these scenarios, LBRF outperforms linear models by at least 5.5%, which shows its adequacy for dealing with nonlinear decision boundaries. The standard deviation is also the smallest or second smallest for LBRF among all classifiers, which suggests better stability. For scenario

2, the best performance is achieved by oblique forest. LBRF's performance, though not the highest, is comparable to the best performing classifier; with their difference being smaller than one standard deviation.

In summary, LBRF performs consistently better than other classifiers considered here when the true signal concentrates on local regions of the whole curve. In scenarios 3 and 4, different local signals have interactions with each other, and the true decision boundaries are not linear. We can see that in these cases the linear models fall behind in prediction accuracy by a significant amount. The second best accuracies are still 3.2% and 5.3% lower than LBRF, which are more than 10 times their standard deviations.

4.3.2 Simulation Results: Variable Importance

In this section, we compare our variable importance measure to some existing methods, namely, penalized linear coefficients, oblique random forest variable importance, rotation forest variable importance, and random forest permutation importance. In Figure 4.4, we plot the variable importance measures for each combination of approach and simulated data scenario. The tuning parameters which we use to calculate these variable importance estimates are chosen to be the ones with the highest cross validation accuracies in Table 4.1.

Comparing Figure 4.4 to Figure 4.2, it can be seen that LBRF successfully finds the region where the true signal lies in all cases. In Scenarios 1, 3, and 4, it pinpoints the location where we place the discriminatory information, and in Scenario 3, it assigns higher importance to those variables where the difference of the amplitude between the two classes is larger.

Compared with LBRF, the variable importance produced by the penalized linear model, random forest, and oblique forest are overly noisy. Rotation forest, on the other hand, seems to provide a better importance estimate than LBRF. In LBRF,

regions that are near the true signals also received a certain amount of importance and this diminishes as you move further away from the signal. We consider this reasonable for functional data since regions near the true signals are also correlated with the true signal. As a result, these regions do possess some predictive information. In comparison, rotation forest uses a randomly selected subset of features and applies PCA on these features to acquire splitting directions. This process does not take the continuity of functional curves into account. Therefore the rotation forest is able to give a clean cut-off at the end points of the true signal region.

4.3.3 Lupus data analysis

In this section we test the performance of LBRF on the Lupus dataset detailed in Project 1. We run each classifier with the different sets of tuning parameters. Figure 4.5 shows different accuracies under different sets of tuning parameters for each method. Table 4.3 reports the best prediction accuracy for each method. Also, Figure 4.6 plots the variable importance produced by each of the classifiers. Comparing Figure 4.6 to Figure 4.1, we can see that the most significant signal occurs around the region from 60°C to 70°C, and neither the penalized linear model nor the oblique forest captured this signal. Random forest does find some true signals, but still only a few variables pop out in its variable importance plot, and a large amount of discriminatory information is not revealed.

4.4 Tables and Figures

Table 4.1: Tuning Parameters

Method	**Tuning Parameters**
glmnet	$\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$
RF	$nodesize \in \{1, 5, 10\}$ $mtry \in \{p/3, \sqrt{p}\}$
obliqueRF	method $\in$ {ridge, svm, log, rand} $mtry \in \{p/3, \sqrt{p}\}$
rotationforest	$K \in \{3, 5, 10\}$ $minsplit \in \{2, 10, 20\}$ $cp \in \{0, 0.01, 0.1\}$
LBRF	$nodesize \in \{1, 5, 10\}$ $L \in \{nodesize/2, \sqrt{nodesize}\}$ $bandwidth \in \{p/2, \sqrt{p}\}$

Table 4.2: Cross validation accuracies of each classifier

Methods	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
glmnet	0.944 (0.015)	0.823 (0.022)	0.689 (0.031)	0.877 (0.024)
glmnet (50 PCs)	0.944 (0.015)	0.826 (0.020)	0.711 (0.025)	0.890 (0.021)
RF	0.918 (0.012)	0.825 (0.015)	0.731 (0.027)	0.895 (0.016)
RF (50 PCs)	0.914 (0.016)	0.835 (0.015)	0.722 (0.024)	0.850 (0.031)
obliqueRF	0.906 (0.020)	0.838 (0.010)	0.690 (0.020)	0.862 (0.023)
rotationForest	0.965 (0.014)	0.832 (0.013)	0.735 (0.023)	0.901 (0.014)
LBRF	0.970 (0.009)	0.831 (0.013)	0.766 (0.021)	0.954 (0.011)

Table 4.3: Cross validation accuracies of each classifier

Method	Accuracy
glmnet	0.867 (0.029)
glmnet (50 PCs)	0.909 (0.027)
RF	0.769 (0.039)
RF (50 PCs)	0.906 (0.028)
obliqueRF	0.904 (0.030)
rotationForest	0.837 (0.035)
LBRF	0.921 (0.026)

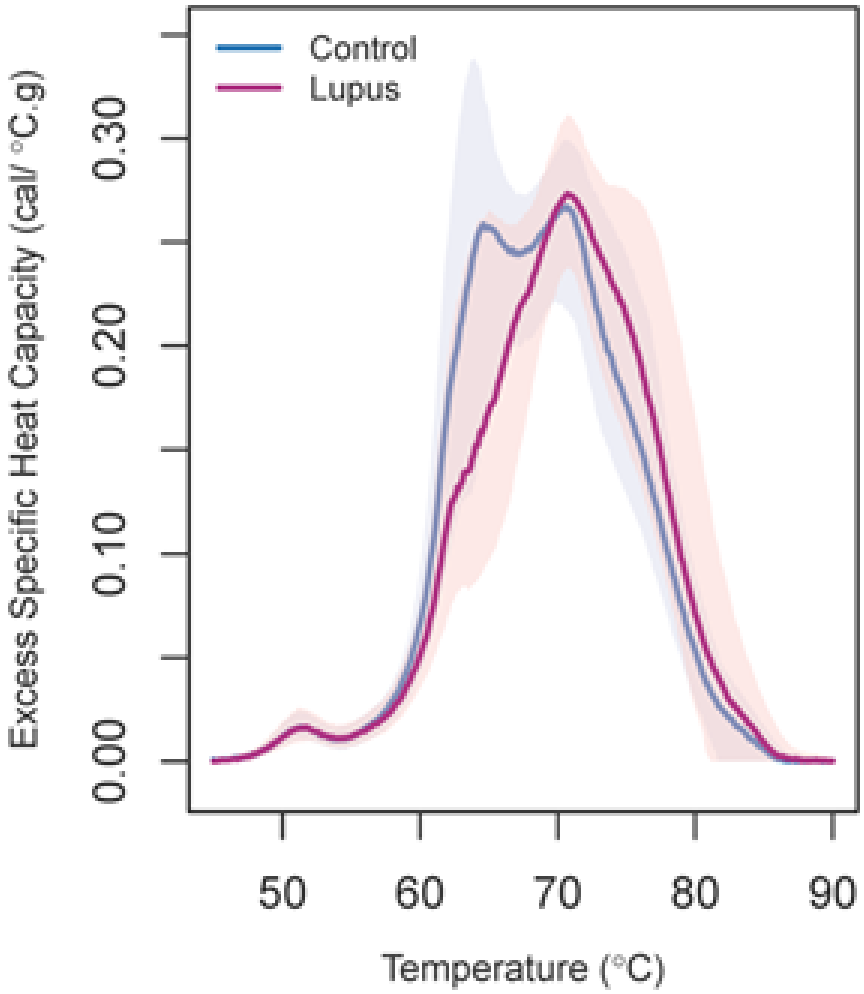

Figure 4.1: The motivating example. Lupus data. An illustration of the functional data classification problem.

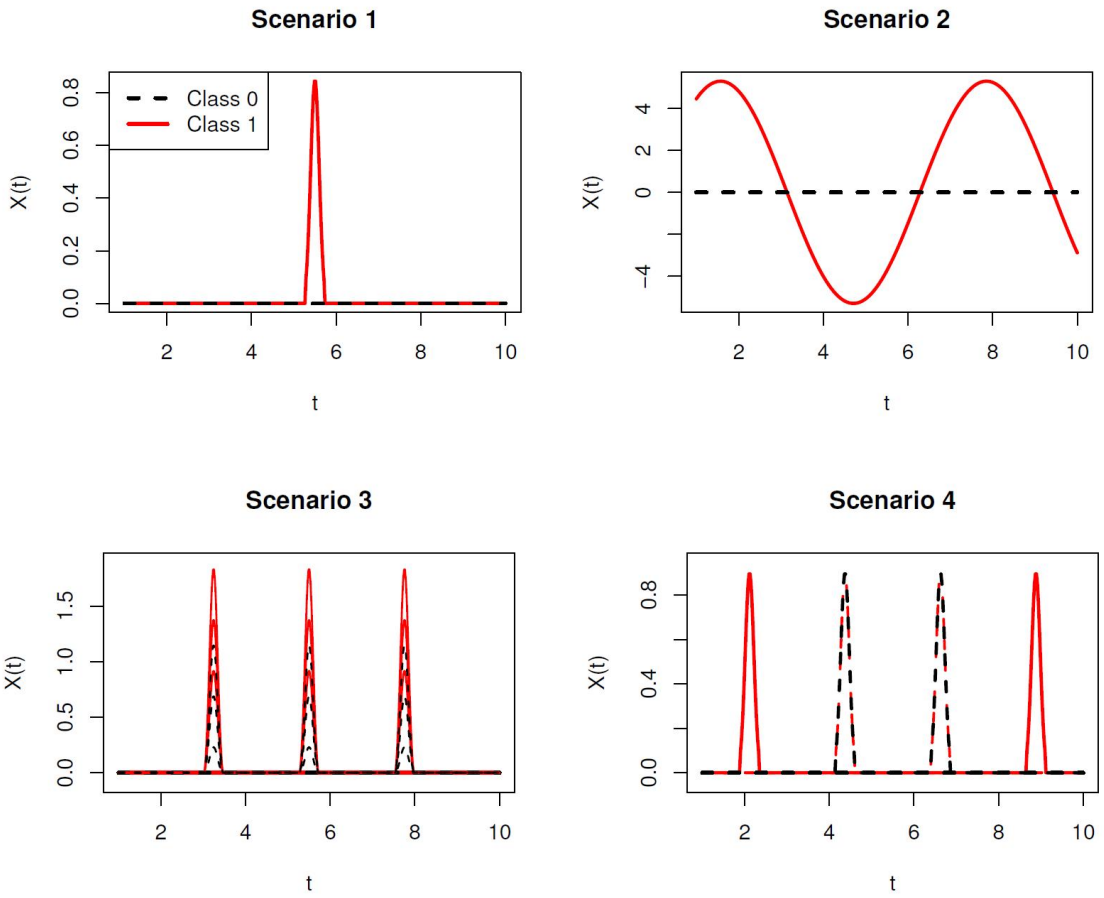

Figure 4.2: Simulated functional curves. The base curve $X_0(t)$ and the noise term Z(t) are not plotted.

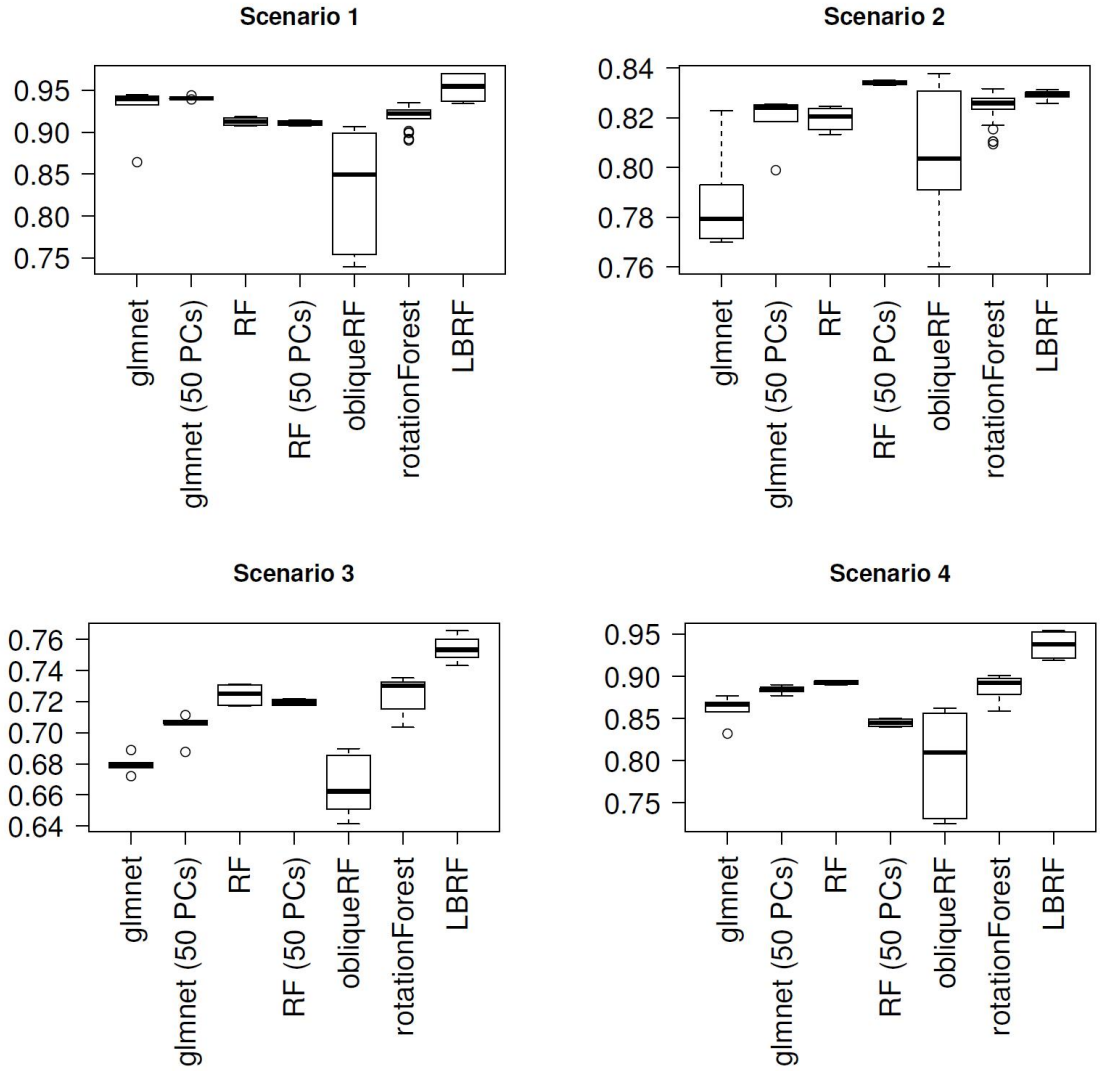

Figure 4.3: Cross validation accuracies of different classifiers for each simulation scenario

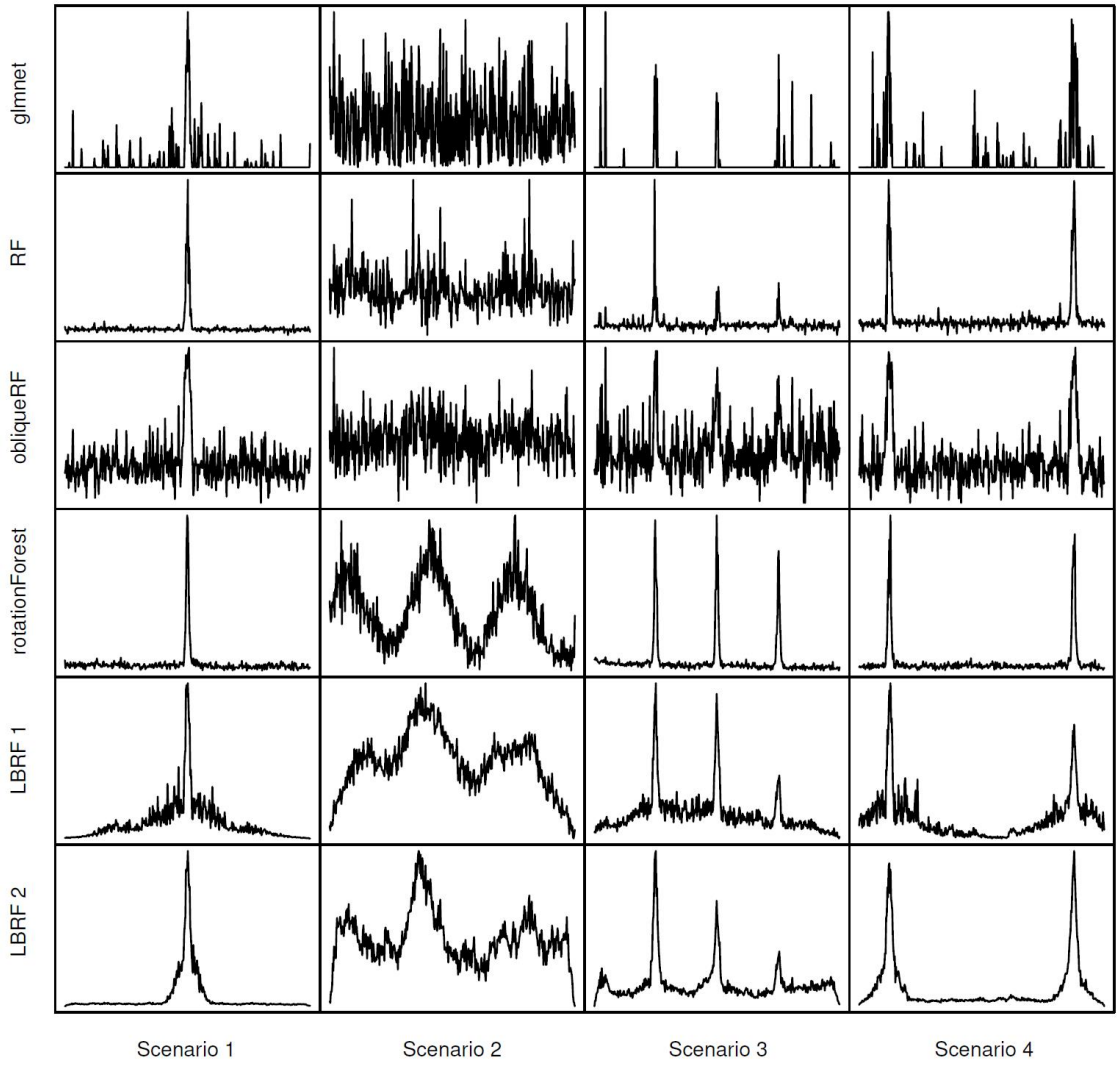

Figure 4.4: Variable importance estimates of the simulated functional datasets. LBRF 1 uses the set of tuning parameters with the highest predicition accuracy, and LBRF 2 has the fixed $bandwidth.max = 0.1p$.

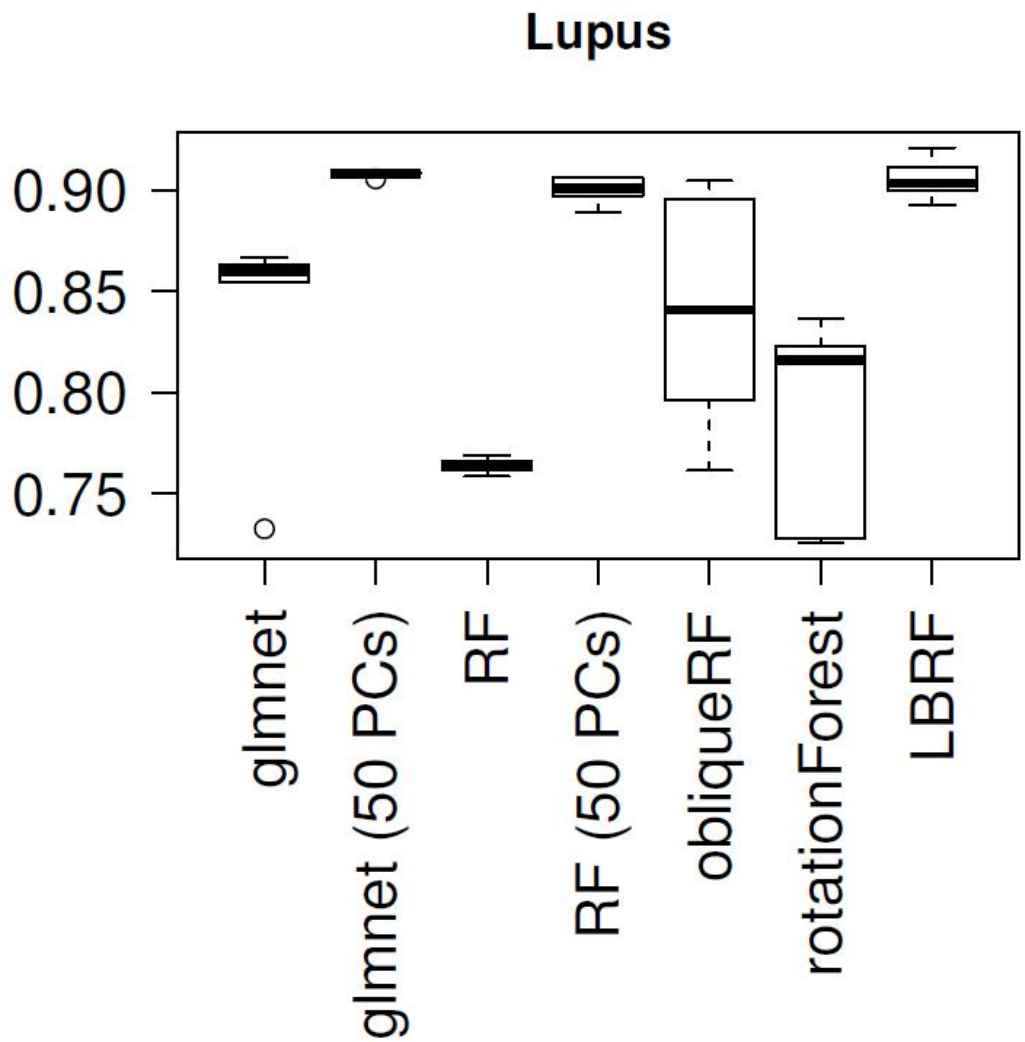

Figure 4.5: Cross validation accuracies of each classifier on the Lupus data.

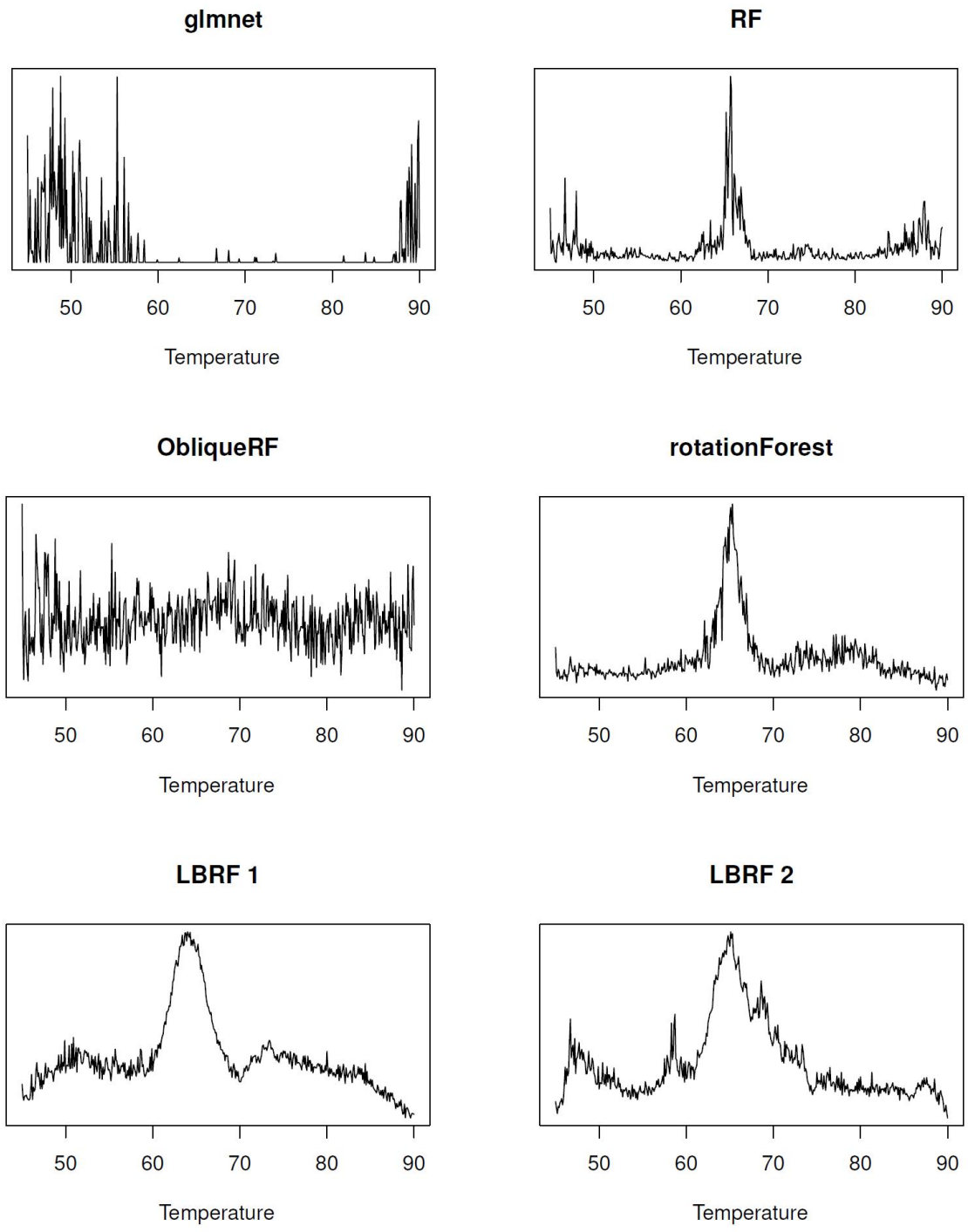

Figure 4.6: Variable importance of Lupus data. LBRF 1 uses the set of tuning parameters with the highest prediction accuracy, and LBRF 2 has the fixed $bandwidth.max = 0.1p$.

CHAPTER 5

CONCLUSION

In Project 1 we explored a parametric approach to functional regression using Lupus DSC data. We implemented three different regression models: 1. Functional linear regression with thermogram as the response variable and disease status as the predictor; 2. Functional generalized linear regression with disease status as the response variable and thermogram as the predictor; and 3. Functional generalized linear regression with disease status as the outcome variable and FPCA scores as the predictors. Here, the FGLM model with FPCA scores gave the highest correct prediction accuracy and most stable regression estimates. One limitation of Project 1 is the assumption that the relationship between the response and predictors is linear. However, this may not always be the case. If this linearity assumption is violated, the FGLM will not perform well. This motivated us to develop a semi-parametric method that is capable of capturing the true underlying relationship - generalized functional partially linear single-index model (GFPL). This model performs well when the linearity assumption is violated. When the relationship is linear, the GFPL performs on par with the standard FGLM. Both projects 1 and 2 take the entire curve into account, however results from these projects indicate that only part or parts of the curve may be truly important. Thus, in Project 3, we develop local basis random forests for functional data, a nonparametric approach to perform classification and calculate variable importance. This method yielded the highest correct classfication in the Lupus data of all methods introduced in this paper and was able to pinpoint the location of the

true discriminatory signal in a variety of simulation settings as well as in the Lupus data. Overall, this dissertation shows that there is promise in using DSC data for predicting disease status for Lupus.

In this dissertation we only examine how our methods perform on the Lupus dataset. However, these methods could be applied to a range of diseases to potentially aid in diagnosis. Also, although this dissertation only explores binary classification, these methods could be extended to the case where there are more than two outcome categories. In future work we would like to examine the performance of the LBRF on derivative curves of functional data; develop methods for handling longitudinal/repeated measures functional data; and explore how using independent component analysis (ICA) in place of PCA would affect the models.

REFERENCES

[1] Garbett NC, Brock GN (2015) Differential scanning calorimetry as a complementary diagnostic tool for the evaluation of biological samples. Biochim Biophys Acta.

[2] Garbett NC, Miller JJ, Jenson AB, Miller DM, Chaires JB (2007) Interrogation of the plasma proteome with differential scanning calorimetry. Clin Chem 53: 2012-2014.

[3] Centers for Disease Control and Prevention (2014) Systemic lupus erythematosus (SLE or lupus).

[4] Mayo Clinic Diseases and Conditions: Lupus.

[5] Ginzler E, Tayar J (2015) Lupus. In: Rheumatology ACo, editor.

[6] Rasmussen A, Sevier S, Kelly JA, Glenn SB, Aberle T, et al. (2011) The lupus family registry and repository. Rheumatology (Oxford) 50: 47-59.

[7] Silverman B, Ramsay J (2005) Functional Data Analysis: Springer.

[8] Febrero-Bande M, de la Fuente MO (2012) Statistical Computing in Functional Data Analysis: The R Package fda.usc. 2012 51: 28.

[9] Fish DJ, Brewood GP, Kim JS, Garbett NC, Chaires JB, et al. (2010) Statistical analysis of plasma thermograms measured by differential scanning calorimetry. Biophys Chem 152: 184-190.

[10] Vega S, Garcia-Gonzalez MA, Lanas A, Velazquez-Campoy A, Abian O (2015) Deconvolution analysis for classifying gastric adenocarcinoma patients based on differential scanning calorimetry serum thermograms. Sci Rep 5: 7988.

[11] Ramsay JO, Hooker G, Graves S (2009) Functional data analysis with R and MATLAB. New York ;: Springer.

[12] Ma S (2016) Estimation and inference in functional single-index models. Annals of the Institute of Statistical Mathematics 68: 181-208.

[13] Carroll RJ, Fan J, Gijbels I, Wand MP (1997) Generalized Partially Linear Single-Index Models. Journal of the American Statistical Association 92: 477-489.

[14] McCullagh P (1984) Generalized linear models. European Journal of Operational Research 16: 285-292.

[15] Cai TT, Hall P (2006) Prediction in functional linear regression. Ann Statist 34: 2159-2179.

[16] Zhou S, Shen X, Wolfe DA (1998) Local Asymptotics for Regression Splines and Confidence Regions. The Annals of Statistics 26: 1760-1782.

[17] Huang JZ (2003) Local Asymptotics for Polynomial Spline Regression. The Annals of Statistics 31: 1600-1635.

[18] de Boor C (2001) A Practical Guide to Splines: Springer New York.

[19] Tripathi G (1999) A matrix extension of the Cauchy-Schwarz inequality. Economics Letters 63: 1-3.

[20] Stone CJ (1986) The Dimensionality Reduction Principle for Generalized Additive Models. The Annals of Statistics 14: 590-606.

[21] Breiman L (2001) Random Forests. Machine Learning 45: 5-32.

[22] Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Machine Learning 63: 3-42.

[23] Zhu R, Zeng D, Kosorok MR (2015) Reinforcement Learning Trees. Journal of the American Statistical Association 110: 1770-1784.

[24] Hall P, Heckman NE (2002) Estimating and Depicting the Structure of a Distribution of Random Functions. Biometrika 89: 145-158.

[25] Zhou J, Wang N-Y, Wang N (2013) Functional Linear Model with Zero-value Coefficient Function at Sub-regions. Statistica Sinica 23: 25-50.

[26] Fan G, Cao J, Wang J (2010) Functional data classification for temporal gene expression data with kernel-induced random forests. 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB): 1-5.

[27] Menze BH, Kelm BM, Splitthoff DN, Koethe U, Hamprecht FA (2011) On Oblique Random Forests. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M, editors. Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 453-469.

[28] Rodriguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: A new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28: 1619-1630.

[29] Spitzner DJ, Marron JS, Essick GK (2003) Mixed-Model Functional ANOVA for Studying Human Tactile Perception. Journal of the American Statistical Association 98: 263-272.

APPENDIX

Appendix 1

Chapter 2 of this dissertation is modified from a paper recently published in Plos One Journal. The paper is entitled "Application and Interpretation of Functional Data Analysis Techniques to Differential Scanning Calorimetry Data from Lupus Patients" and authored by S.Kendrick, Q. Zheng, N. Garbett, and G. Brock.

CURRICULUM VITA

NAME: Sarah Kendrick

ADDRESS: Department of Biostatistics and Bioinformatics
University of Louisville
Louisville, KY 40292

EDUCATION: B.A.,
Wittenberg University, 2010
M.S.,
University of Louisville, 2013

PUBLICATIONS: Kendrick, S. Wu, D. (2015).
Simulation study for the Lead Time in Cancer Screening when Human Life Time is a Competing Risk.
Journal of Biometrics and Biostatistics.

Kendrick, S. Zheng, Q. Garbett, N. Brock, G. (2017).
Application and Interpretation of Functional Data Analysis Techniques to Differential Scanning Calorimetry Data from Lupus Patients.
Plos One.

MEMBERSHIP AND

LEADERSHIP: American Statistical Association Member

[2013−Present]

Biostatistics Club President

[2015, 2016]