The Pennsylvania State University

The Graduate School

FUNCTIONAL MANIFOLD DATA ANALYSIS

A Dissertation in

Department of Statistics

by

Hyun Bin Kang

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

August 2018

The dissertation of Hyun Bin Kang was reviewed and approved* by the following:

Matthew Reimherr
Assistant Professor of Statistics
Dissertation Adviser and Chair of Committee

Runze Li
Eberly Family Chair in Statistics

Bing Li
Professor of Statistics

Mark Shriver
Professor of Anthropology

Naomi S. Altman
Professor of Statistics

*Signatures are on file in the Graduate School.

Abstract

With rapid advances in data collection technologies, many scientific fields are now obtaining more detailed, complex, and structured data. Utilizing such structures to extract more information is increasingly common in fields such as biology, anthropology, forensic science, and meteorology. A great deal of modern statistical work focuses on developing tools for handling such data. Classic statistical tools such as univariate or multivariate methods are often not suitable for such data and in some situations, and in some cases applying them is not even possible because of data structure.

In this dissertation, we propose *Functional Manifold Data Analysis (FMDA)*, a subbranch of Functional Data Analysis (FDA) which often extracts additional information contained in the data structure, to deal with such modern complicated data. FDA is a rapidly developing area of statistics for data which can be naturally viewed as a smooth curves or functions. In FDA, the fundamental statistical unit is now function or shape, not the vector of measurements, and the inherent smoothness in the data can be exploited to achieve greater statistical efficiency than typical multivariate methods. In particular, we present an inferential framework when one of the variables being analyzed is a manifold, and thus we assume we have as many manifolds as we have units. To achieve this, we utilize deformation maps from shape analysis and dimension reduction techniques from manifold learning. In doing so, we are able to represent each manifold as a deformation map, which then can be analyzed using functional data methods. Currently, shape analysis methods that go beyond an analysis of landmarks is a very active area of research, but to date, little has been done in terms of shape-on-scalar regression, thus this dissertation will open up exciting avenues for both shape and functional data analysis.

To understand how the scalar covariates affect the manifolds, we propose a

manifold-on-scalar regression, which is an extension of function-on-scalar regression in FDA. Different algorithms for estimating the parameter functions in the manifold-on-scalar regression are presented and discussed. The optimality of parameter estimates for function-on-scalar regression over complex domains is also established by finding the minimax lower bounds on the estimation rate and proposing a minimax optimal estimator whose upper bounds match the developed lower bounds.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Functional Data Analysis

Functional Data Analysis (FDA) is a branch of statistics that deals with theories and analysis of the information on functions, curves, shapes, images, or objects. Ramsay and Dalzell (1991) has provided the foundation of FDA and introduced the analysis of infinite dimensional processes. Functional data are intrinsically infinite dimensional, but they are measured with discrete points. Instead of treating these measurements as a set of vectors, FDA considers them as continuous functions and models them in function space. Ramsay and Silverman (2005) summarized the aims of functional data analysis as 1) to present data to in ways that help further analysis, 2) to display data in ways to highlight various characteristics, 3) to understand pattern and variation among data, and 4) to explain variation in dependent variable in association with independent variable information.

FDA has seen a precipitous growth in recent years, and its application can be

found in many different fields including genetics for understanding human growth patterns (Chen and Müller, 2012; Verzelen *et al.*, 2012) or gene expression (Tang and Müller, 2009), finance for credit card transaction volumes (Kokoszka and Reimherr, 2012), geoscience for finding geomagnetic activity patterns (Gromenko and Kokoszka, 2013), and neuroimaging (Reiss *et al.*, 2011; Zipunnikov *et al.*, 2011), to name only a few. Some FDA methods and applications to real data has been detailed in the textbook by Ramsay and Silverman (2007), and Horváth and Kokoszka (2012) presented more recent methodologies and theories with more focus on inference with applications to geosciences, finance, economics and biology. As FDA becomes a common tool in Statistics, Kokoszka and Reimherr (2017) published an introductory textbook to the field.

1.1.1.1 Basis Expansion

Classic functional data analysis involves the analysis of time series data, the functions over time. A famous example of functional data analysis is the Canadian weather data (Ramsay and Silverman, 2005) in Figure 1.1. At thirty-five weather stations across Canada, the temperature is measured everyday over the course of a year, and the mean monthly temperature is shown in Figure 1.1. The lines can be considered as the temperature functions over time, i.e. $\{X_n(t) : n = 1, \cdots, 35, t \in \mathcal{T}\}$ where $\mathcal{T}$ is from January to December, and each point on the plot X_{nj} is considered as $X_n(t_j) + \epsilon_{nj}$, the function evaluated at one point with measurement errors ϵ_{nj}.

A common way to represent this data is using nonparametric smoothing methods. We expand the functions using basis functions. In the systematic review of 84 FDA application articles, Ullah and Finch (2013) revealed that 72 studies (85.7%) provided information about the type of smoothing techniques used, with B-spline

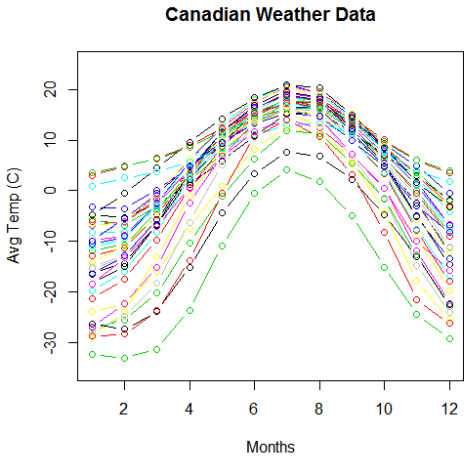

Figure 1.1: Raw data for mean monthly temperatures at thirty-five Canadian weather stations.

smoothing being the most popular and Fourier-basis smoothing being the second most popular. Fourier basis functions are sine and cosine functions of increasing frequency, so they are especially useful for periodic data like the Canadian weather data. Examples of Fourier basis functions are shown in the left of Figure 1.2. We can represent the temperature functions using the Fourier basis $B_m(t)$:

$$X_n(t) \approx \sum_{m=1}^{M} c_{nm} B_m(t), \quad 1 \leq n \leq 35, \quad t \in \mathcal{T}.$$

The coefficients c_{nm} are estimated by minimizing

$$\sum_{n=1}^{35} \sum_{j=1}^{12} \left(X_{nj} - \sum_{m=1}^{M} c_{nm} B_m(t_j) \right)^2 + \lambda \int L \left[\left(\sum_{m=1}^{M} c_{nm} B_m(t_j) \right] \right)^2 dt$$

where L is an operator that measures the roughness of functions, and λ is the smoothing parameter which governs the trade-off between the fit to the measurements and the smoothness of resulting functional objects. The choice of λ is usually

chosen using the generalized cross validation (Golub *et al.*, 1979).

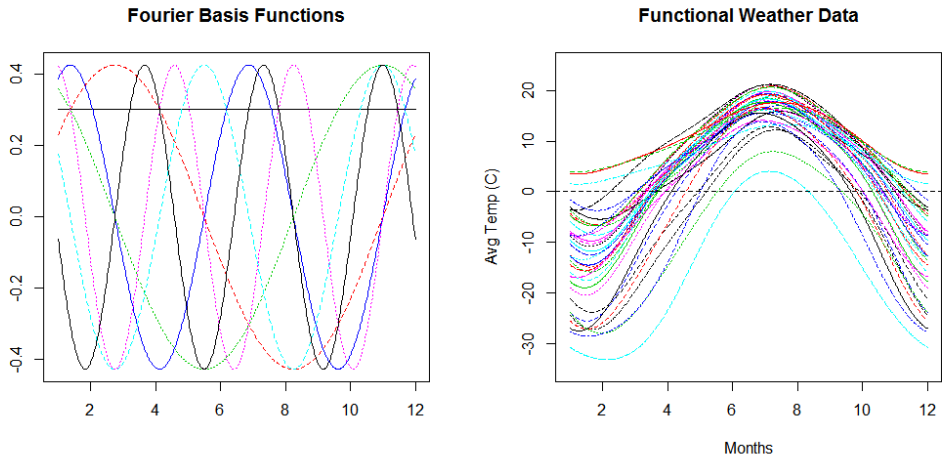

Figure 1.2: Temperature functional objects are shown on the right and these are expanded with Fourier basis functions on the left.

FDA methods exploit what Ramsay and Silverman (2005) termed replication and regularization. In particular, unlike classic nonparametric smoothing methods, the data usually consist of as many functions as there are statistical units, while the inherent smoothness in the data/parameters can be exploited to achieve greater statistical efficiency than typical multivariate methods. By treating the data as functions and by making a weak assumption that those functions are smooth, we are able to exploit the smoothness and analyze the characteristics of functional data such as the uses of derivatives in modeling (Ramsay, 2006). For the Canadian weather data too, it is possible to check the rate of temperature change by taking one derivative of the temperature functions. Another example would be the growth curves of children which are height measurements over time (Kokoszka and Reimherr, 2017) and by considering them as functions, it becomes possible to analyze the rate of growth of children.

The functional objects $X_n(t)$ are considered as the realizations of a random

function $X(t)$, and its mean and covariance functions are defined as

$$\mu(t) = E\left[X(t)\right],$$

$$\Sigma(t,s) = E\left[\left(X(t) - \mu(t)\right)\left(X(s) - \mu(s)\right)\right],$$

and these two are estimated by

$$\hat{\mu}(t_j) = \frac{1}{N}\sum_{n=1}^{N} X_n(t_j),$$

$$\hat{\Sigma}(t_j,t_k) = \frac{1}{N}\sum_{n=1}^{N}\left(X(t_j) - \hat{\mu}(t_j)\right)\left(X(t_k) - \hat{\mu}(t_k)\right).$$

1.1.1.2 Functional Principal Component Analysis

Functional Principal Component Analysis (FPCA) is an extension of multivariate PCA to the functional case. It played major role in the early FDA literature, and it is still well-studied and considered important. As with multivariate PCA, FPCA explores the covariance structure of the functional objects, and it identifies functional principal components that explain the most variability of a sample of curves. In addition to providing such information, FPCA also allows the functions to be expanded with its orthonormal eigenfunctions, analogous to the orthogonal eigenvectors in multivariate case. Using the expansion with eigenfunctions, we are able to reduce down the dimension of functions, as the multivariate PCA is used for linear dimension reduction. Previously functional objects are expanded using M basis functions, and therefore there were M coefficients for each function, but after FPCA, the functions can be represented with $J \leq M$ number of eigenfunctions.

Mercer's theorem gives that the spectral decomposition of Σ as

$$\Sigma(t,s) = \sum_{k=1}^{\infty} \tau_k v_k(t) v_k(s)$$

where τ_k's are eigenvalues in descending order and $v_k(t)$'s are the corresponding eigenfunctions. Karhunen and Loéve (Karhunen, 1946; Loéve, 1946) also found independently that every square integrable function X can be represented as

$$X(t) = \mu(t) + \sum_{k=1}^{\infty} \tau_k v_k(t)$$

and this expansion is called *Karhunen-Loéve expansion*.

The estimation of the eigenvalues τ_k and eigenfunctions $v_k(t)$ is done by estimating the covariance operator, evaluating it on a grid, and conducting a matrix spectral decomposition on the grid-evaluated covariance. The convergence of the estimated eigencomponents is obtained by integrating the results on the convergence of the covariance estimates that are attained under regularity conditions with perturbation theory (Kato, 1966).

FPCA also facilitates functional principal component regression. By projecting functional responses or predictors to their first few principal components, it becomes possible to apply regression models with those vector responses or predictors. FPCA is also useful for classification and clustering of functional data since it is a crucial dimension reduction tool in FDA.

1.1.1.3 Functional Regression

In order to understand the relationship between functional data and other variables, functional regression models are developed. Depending on whether the functional data are the predictors or the responses, the model structure changes.

Scalar-on-function regression is when the functional data are included as the predictors or the covariates and the response variable is scalar.

$$Y_n = \int_{\mathcal{T}} \beta(t) X_n(t) dt + \epsilon_n.$$

Function-on-scalar regression is when the functional data are included as the response while the predictors are scalar.

$$Y_n(t) = \sum_{p=1}^{P} x_{np} \beta_p(t) + \epsilon_n(t).$$

Function-on-function regression also follows the same rule. This is when both the response and the predictor are functional data.

$$Y_n(t) = \int_{\mathcal{T}} \beta(t,s) X_n(s) ds + \epsilon_n(t).$$

Concurrent linear model is a special case of when both the response and the predictor are functional data. The model looks like

$$Y_n(t) = \beta_0(t) + \beta_1(t) X_n(t) + \epsilon_n(t).$$

In this case, Y_n at time t_j is affected only by X_n at the same time t_j whereas in function-on-function regression model, X_n at time t_j can affect Y_n at all other t_k for $k \neq j$ too.

It is also possible to have generalized linear model version of the models above. With scalar-on-function case, it is easy to assume that the scalar response have error distribution other than normal distribution. However, with functional response, it is not conceptually intuitive. Current version dictates that $Y_n(t)$ satisfies a generalized linear model with the scalar predictor at each point t_j with the same

density across t.

There are also random effects models and mixed effect models, but we are not going to dwell on those. In many cases, FPCA is utilized to alleviate the curse of dimensionality for the nonparametric procedures.

In this dissertation, we will propose an extension of function-on-scalar regression, which we call as *manifold-on-scalar regression* where the functional response can be considered as manifolds that are sitting in $\mathbb{R}^D$ with $D > 1$ and thus is a D-dimensional vector of functions.

1.1.1.4 Recent Advances in Functional Data Analysis

FDA has seen a precipitous growth in recent years, due in part to the numerous complex data that have emerged. With the growth and advance of modern data collection technology, the need to understand the complex structures of data has been increased. Wang *et al.* (2016) termed the functional data which are multivariate, correlated, high-dimensional, and part of complex objects as *next-generation functional data*.

An example of the next-generation functional data is the acoustic phonetic data analyzed by Pigoli *et al.* (2018). They pre-process the samples of sound recordings of different words in different languages pronounced by different speakers to make them decibel functions in time and frequency domain, $\{X_n(\omega, t) : n = 1, \cdots, N, \omega \in [0, \Omega], t \in [0, T]\}$. In classic FDA, the functions are defined on a single one-dimensional domain, but Pigoli *et al.* (2018) move on to a two-dimensional domain of time and frequency.

Examples of next-generation functional data can also be found in brain and neuroimaging. Neurologists aim to map the neuron activity of the human brain and measure signals over the whole brain. These signals can be viewed as functions

over the brain.

In recent years there has also been an increased interest in exploring how manifold structures impact functional data techniques. Chen and Müller (2012) consider extending functional data techniques to data that are all lying on a single manifold. Elhamifar and Vidal (2011) consider clusters of functional data with each cluster lying on a different manifold. Ellingson *et al.* (2013) consider mean estimation from functional data all lying on a common manifold. Dimeglio *et al.* (2014) also try to find a template function using manifold embedding, considering observed functions as variables with values on a single manifold. Additionally, in 2014, SAMSI had a working group on *Data Analysis on Hilbert Manifolds and their Applications* (Bunea *et al.*, 2014), and Lila *et al.* (2016) provide a smooth principal component analysis algorithm for functions on a two-dimensional manifold. Ettinger *et al.* (2016) map the internal carotid artery on a planar domain, which is also a manifold that is homeomorphic to a cylinder. The recent paper by Lila and Aston (2017) proposed a framework for textured data on two-dimensional manifold domain where domains are subject to variability from sample to sample with an application to medical imaging. They consider the thickness of brain as a function over the whole brain, meaning that they took the brain manifold as a domain, and they also allow that the domain is subject to variability from sample to sample.

FDA tools are also widely used in a very recently developing field called Object Oriented Data Analysis (Marron and Alonso, 2014; Patrangenaru and Ellingson, 2015) which concerns the statistical analysis of samples of complex objects such as shapes, images, and trees. Wang and Marron (2007) connected FDA to Object Oriented Data Analysis (OODA) and introduced OODA for a tree-structured data with an application to blood vessel dataset. Aydin *et al.* (2009) then developed

a principal component analysis for tree-structured data which is a standard FDA technique. A nonparametric regression model with tree-structured objects as responses has also been developed by Wang *et al.* (2012). Further analysis on tree object data has been developed (Shen *et al.*, 2014a; Skwerer *et al.*, 2014b) but there are few works concerning manifolds in OODA context (Marron, 2014). Therefore, we believe our framework in Chapter 2 will be a valuable asset to OODA literature.

1.1.2 Reproducing Kernel Hilbert Space

Reproducing Kernel Hilbert Space (RKHS) refers to the special case of Hilbert space where there exists a kernel that reproduces every function in the space, meaning that when $X(u)$ is in RKHS, then for any u in the set $\mathcal{U}$ where the function is defined on, the evaluation of X at u can be performed by taking an inner product with a kernel k, i.e. $\forall u \in \mathcal{U}, \forall X \in \mathbb{H}, \langle X, k(\cdot, u)\rangle_{\mathbb{H}} = X(u)$.

1.1.2.1 Benefits of using RKHS in FDA

RKHSs provide a variety of benefits for functional data analysis. The first is that the kernel can be tailored to reflect certain beliefs or assumptions about the parameters, e.g. smoothness or periodicity. The second is that the eigenfunctions of the kernel can be used as a basis for approximating functional observations and/or parameters estimates, though the reproducing property can also be used to obtain parameter estimates. Lastly, commonly used spaces, such as Sobolev spaces, as well as estimation techniques such smoothing splines can naturally be viewed in an RKHS framework.

Let $\mathcal{U} \subset \mathbb{R}^D$ be a compact d-dimensional manifold with $d \leq D$. A kernel function, $K : \mathcal{U} \otimes \mathcal{U} \to \mathbb{R}^+$, is a bivariate function that is symmetric, positive definite, and continuous. There is a one-to-one correspondence between RKHS

and kernel functions. One can generate the RKHS from K by the following. Let $L^2(\mathcal{U})$ denote the space of all square integrable functions from $\mathcal{U}$ to $\mathbb{R}$. We will write $L^2(\mathcal{U})$ as L^2 for simplicity. Furthermore, any norm $\|\cdot\|$ or inner product $\langle\cdot,\cdot\rangle$ written without subscript is understood to be with respect to L^2. By Mercer's theorem we can write

$$K(u,u') = \sum_{j=1}^{\infty} \tau_j v_j(u) v_j(u'),$$

where $v_j \in L^2$ are orthonormal and $\{\tau_j\}$ is a positive, decreasing, summable sequence. We can define a set

$$A_K = \left\{ f \in L^2 : \sum_{j=1}^{\infty} \frac{\langle f, v_k\rangle^2}{\tau_j} < \infty \right\}.$$

We can then define an inner product on A_K as

$$\langle f, g\rangle_{\mathbb{K}} = \sum \tau_j^{-1} \langle f, v_j\rangle \langle g, v_k\rangle.$$

Then we can say that $\mathbb{K}$ is an RKHS with kernel K.

1.1.2.2 Approximating Eigenfunctions of RKHS

Since in most cases it is not possible to get the eigenfunctions $v_j(u)$ of RKHS, we need to approximate them. Basically we want to find τ_j and $v_j(u)$ such that

$$\tau_j v_j(u) = \int_{\mathcal{U}} K(u,u') v_j(u') du'$$

for $\forall u \in \mathcal{U}$ and $j \geq 1$. The eigenfunctions v_j need to be orthonormal in $L^2(\mathcal{U})$, meaning that $\int_{\mathcal{U}} v_j(u) v_k(u) du = 1$ only if $j = k$ and $\int_{\mathcal{U}} v_j(u) v_k(u) du = 0$ otherwise, and v_j also need to be orthogonal in $\mathbb{K}$ with $\|v_j\|_{\mathbb{K}} = \frac{1}{\tau_j}$.

Pazouki and Schaback (2011) present an algorithm to approximate these v_j's. Assume that there are sufficiently large number of points $\{u_1, \cdots, u_M\}$ so that the integration over $\mathcal{U}$ can be approximated by

$$\int_{\mathcal{U}} f(u)du \approx \sum_{m=1}^{M} w_m f(u_m).$$

Then

$$\tau_j v_j(u_l) = \int_{\mathcal{U}} K(u_l, u)v_j(u)du$$

becomes

$$\tau_j \sqrt{w_l} v_j(u_l) \approx \sum_{m=1}^{M} \sqrt{w_l} K(u_l, u_m)\sqrt{w_m}\sqrt{w_m} v_j(u_m).$$

By noting $\sqrt{w_l} v_j(u_l) = \phi_l^{(j)}$, $\sqrt{w_m} v_j(u_m) = \phi_m^{(j)}$, and $\sqrt{w_l} K(u_l, u_m)\sqrt{w_m} = b_{lm}$, the equation becomes a discrete eigenvalue problem:

$$\tau_j \phi_l^{(j)} = \sum_{m=1}^{M} b_{lm} \phi_m^{(j)}.$$

Through this algorithm, we can approximate τ_j and v_j, and we can expand functions using these eigenfunctions v_j as bases.

1.2 Contribution

With rapid advances in data collection technologies, many scientific areas are faced with the challenge of extracting information from large, complex, and highly structured data sets. A great deal of modern statistical work focuses on developing tools for handling such complex objects. In this dissertation, we present a new subfield of Functional Data Analysis (FDA) that concerns the statistical analysis of samples where one or more variables measured on each unit is a manifold. We call this

Functional Manifold Data Analysis (FMDA).

FMDA is motivated by the high-resolution 3D facial imaging data collected from ADAPT study (Anthropology, DNA, and the Appearance and Perception of Traits), an ongoing study at the Pennsylvania State University. Investigators of ADAPT collected 3D facial images, alongside genetic information, from admixed populations in the US, Brazil, and Cape Verde (Claes *et al.*, 2014a,b) in order to understand the variation of human facial structures.

Understanding the architecture of human facial diversity has been of long standing interest in a variety of fields including anthropology and forensic sciences. New methodologies have been developed as more sophisticated bioimaging technologies enable researchers to collect richer 3D facial images. There has been extensive research on 3D facial analysis, but the methodologies have been primarily developed in computer science, electrical engineering, and computer vision for face and facial expression recognition (e.g Turk and Pentland, 1991; Ahonen *et al.*, 2006; Jain and Li, 2011; Taigman *et al.*, 2014; Huang *et al.*, 2014). There is a more limited literature on 3D facial analysis using a statistical framework, and the few existing methods are primarily concerned with classification or estimation based on facial features, not on understanding the influence of different covariates on the 3D faces themselves. Huang *et al.* (2014) introduce a local descriptor multi-modal (2D and 3D) for facial gender and ethnicity classification. Xia *et al.* (2013) adopt machine learning techniques to find relationships between gender and facial asymmetry. Xia *et al.* (2014) examine age effects using a random forest-based regression, but the regression uses features of local shape deformation between facial curves, captured by *Dense Scalar Fields* based on Riemannian shape analysis (Drira *et al.*, 2012). Kurtek and Drira (2015) provide a statistical shape analysis framework for 3D faces which allows comparison, deformation, and expression and identity classifi-

cation, but there has not yet been a corresponding regression method developed that directly takes the whole 3D faces as variables.

Instead, we propose a novel functional data approach to analyze 3D faces, which are viewed as smooth manifolds. By constructing 3D facial functional objects, we can utilize existing functional data analysis tools. Our goal is to build statistical models that elucidate how different covariates affect patterns seen in different faces. However, we do not reduce the faces down to a few quantitative traits; instead we exploit inherently smooth structures in the face so that they can be analyzed as a whole. This provides a unique approach that, for example, takes whole manifolds as variables in a regression model. As a contrast, Fletcher (2013) provides a geodesic regression that takes points on a manifold as the response. Cao *et al.* (2014) explore a shape regression approach that uses a few landmarks on shapes/manifolds. Yang and Dunson (2016) present a Bayesian manifold regression that assumes that the data lie in a subspace that is a single locally-Euclidean compact Riemannian manifold.

FMDA has a unique stance in FDA literature too. Although there are increasing interest towards associating functional data with manifolds, all data is assumed to lie on a single manifold or a small number of manifolds in most previous FDA work. In this dissertation, however, we assume that each unit is its own manifold, meaning that we have as many manifolds as we have units. High-dimensional 3D imaging is becoming more common in fields such as biology, kinesiology, engineering, and anthropology. In many of these cases, each image is actually a surface sitting in 3D space, i.e. a manifold. However, from image to image, the manifold changes and one cannot assume that all images lie on the same manifold.

Our approach utilizes deformation maps from shape analysis and dimension reduction techniques from manifold learning, which allows us to represent each

face/manifold as a function, which can then be analyzed using FDA techniques. Currently, shape analysis methods that go beyond an analysis of landmarks is a very active area of research; our hope is that building a connection with FDA will open up exciting avenues for both shape and functional data analysis, while providing powerful and flexible statistical tools.

We believe that these techniques will prove useful to a variety of applications. As part of the big data revolution occurring in the sciences, many types of high-frequency or high-resolution data are being collected. Data which include samples of manifolds will become increasingly common, especially in biomedical imaging. FDA tools naturally exploit smoothness and we thus believe they will be useful for analyzing data involving manifolds.

1.3 Organization

The dissertation is organized as follows. In Chapter 2, we first present a novel framework to embed a sample of manifolds in a real separable Hilbert space. Then we introduce tools to analyze these functional manifolds such as a 2-step functional principal component analysis (FPCA) algorithm and a manifold-on-scalar regression model, which is an extension of function-on-scalar regression. We show how to get a least square estimator as well as a penalized least square estimator of parameter function and describe the testing methods. Then we apply these methods to the ADAPT data of 3D facial images.

In Chapter 3, we present three algorithms for parameter estimation in manifold-on-scalar regression in RKHS. The first one is a principal component regression method, the second one is also a principal component regression but with smoothness imposed on the covariance operator, and the third is a penalized regression method. Their predictive performance are compared by 10-fold cross validation on

the ADAPT data.

In Chapter 4, the optimality of parameter estimates for function-on-scalar regression over complex domains is established. The minimax lower bounds on the estimation rate are found and a minimax optimal estimator whose upper bounds match the developed lower bounds is proposed. Through simulations we show how the parameter estimation error converges in relation to the sample size and the number of observed measurements per curve.

Chapter 2

Functional Manifold Data Analysis

2.1 Methodology

In this Section we present our approach for handling random samples of manifolds. Our primary goal is to lay the foundation for analyzing such data using FDA tools. To accomplish this, we use tools from shape analysis so that each manifold can be associated with a particular deformation map, while we use tools from manifold learning and FDA to carry out the described computations. A visual outline of our approach can be found in Figure 2.1.

In Section 2.1.1 we introduce a statistical framework to embed a sample of manifolds into a real separable Hilbert space, resulting in a sample of functional objects. We then present a 2-step Functional Principal Component Analysis (FPCA) algorithm in Section 2.1.2. In Section 2.1.3, we discuss a manifold-on-scalar regression model and hypothesis testing methods for its coefficient functions.

2.1.1 Algorithm

In order to ensure the manifolds are comparable and that our algorithm can be applied, we make the following assumptions.

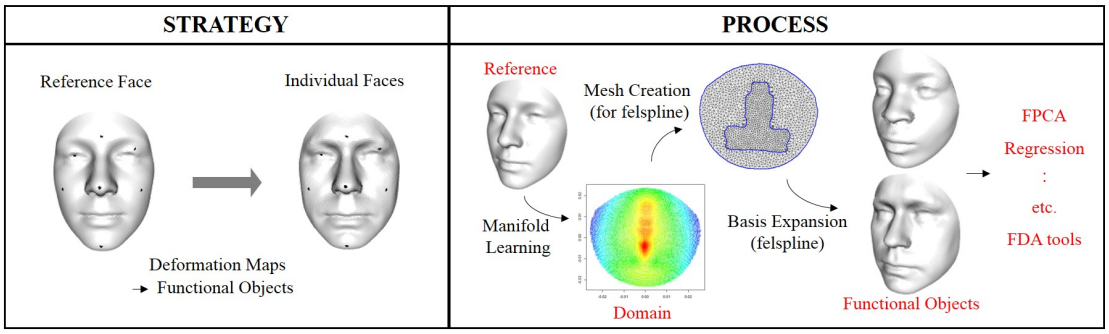

Figure 2.1: The strategy and process of Functional Manifold Data Analysis (FMDA) are presented. Individual faces are considered as deformations maps of a reference face, and using this reference face as domain, we construct functional data objects. After that, we can apply functional principal component analysis (FPCA), functional regression, or other FDA tools to analyze them.

Assumption 2.1.1. *Let $\mathcal{Y}_1, \cdots, \mathcal{Y}_N$ be a random sample of manifolds. We assume that, with probability one,*

1. *each $\mathcal{Y}_n$ is a compact d-dimensional manifold that is a subset of $\mathbb{R}^D$ with $d < D$,*

2. *there exists a nonrandom compact d-dimensional C^1 Riemannian manifold $\mathcal{M}_0$ such that each $\mathcal{Y}_n$ is homeomorphic to $\mathcal{M}_0$,*

3. *there exists an atlas for $\mathcal{M}_0$ with a single coordinate chart $\{(\mathcal{M}_0, \psi)\}$ where for any open set $U \subset \mathcal{M}_0$, $\psi : U \to \psi(U) \subset \mathbb{R}^d$ and $\mathbf{M}_0 \triangleq \psi(\mathcal{M}_0)$,*

4. *to each manifold $\mathcal{Y}_n$, there exists a function $\mathbf{Y}_n : \mathbf{M}_0 \to \mathbb{R}^D$ such that $\mathbf{Y}_n(\mathbf{M}_0) = \mathcal{Y}_n$, up to possibly a set of measure 0,*

5. *the functions $\mathbf{Y}_n$ are elements of $L^2(\mathbf{M}_0)$ with probability one, i.e. $\int_{\mathbf{M}_0} \mathbf{Y}_n^\top(m)\mathbf{Y}_n(m)dm < \infty$.*

The first assumption states that the sample consists of manifolds that are in the same ambient space, $\mathbb{R}^D$. This can be generalized to other spaces, but we do

not pursue that here given the scope of our intended applications. The second guarantees that the manifolds are comparable by assuming that they can all be parametrized by a common manifold, $\mathcal{M}_0$. This manifold is assumed to be C^1 and Riemannian so that integration over the manifold is well defined (Lee, 2003). The third assumption lets us apply manifold learning methods to "unfold" $\mathcal{M}_0$ into the simpler set $\mathbf{M}_0$. This is primarily for computational convenience, as $\mathbf{M}_0$ is an easier domain to work with. If the third assumption does not hold, then the manifold, $\mathcal{M}_0$, cannot be mapped to a set in $\mathbb{R}^d$ without tearing it in some way. Such an assumption is reasonable for our facial applications, but, for example Ettinger *et al.* (2016) utilize FDA methods for data measured on the internal carotid artery, which is homeomorphic to a cylinder and thus would violate this assumption. The fourth assumption simply allows us to identify the manifolds as functions, which are commonly referred to as deformation maps in shape analysis, while the fifth assumption allows us to view those functions as elements of a Hilbert space.

At the heart of our methodology is the view that each manifold can be identified with a function, and then properties such as smoothness can be defined and exploited by utilizing these functions. The major difference between our setting and traditional FDA is that the domain, $\mathbf{M}_0$, is not observed and must therefore be constructed. The framework to construct $\mathbf{Y}_n : \mathbf{M}_0 \to \mathbb{R}^D$ from $\mathcal{Y}_n$ is summarized below.

1. Identify a reference manifold $\mathcal{M}_0$.
2. Embed $\mathcal{M}_0$ into a closed bounded connected region of $\mathbb{R}^d$ to construct $\mathbf{M}_0$.
3. Construct basis functions from $\mathbf{M}_0$ to $\mathbb{R}^D$ and express $\mathbf{Y}_n$ as a linear combination of these functions.

We now discuss each of the steps above. We assume that the raw data is of

the form $\{\mathrm{y}_{npq} : n = 1, \cdots, N; p = 1, \cdots, P; q = 1, \cdots, D\}$, which consists of P D-dimensional points observed on manifolds $\mathcal{Y}_1, \cdots, \mathcal{Y}_N$. Notice that for 3D facial data, D equals 3, P represents the number of points observed per face, and N is the number of collected faces. We assume that each manifold is ultra-densely sampled, and thus can be completely reconstructed with almost no error, which is a common assumption in *Dense Functional Data Analysis* (Zhang and Wang, 2016). With the development of data collection technologies, these types of data are increasingly common.

For the first step, the reference manifold $\mathcal{M}_0$ can be taken from an external source, such as previous literature or a previously constructed library of objects, one of the manifolds in the sample, or an average from the sample.

Once $\mathcal{M}_0$ is identified, we use manifold learning techniques on $\mathcal{M}_0$ to find $\mathbf{M}_0$, the embedding of $\mathcal{M}_0$ in to $\mathbb{R}^d$. The resulting points will be denoted as $\{\mathrm{m}_{pq} : p = 1, \cdots, P; q = 1, \cdots, d\}$, P. Manifold learning has been a very active area of research, and there are a number of popular methods for carrying out this step, including Isomap (Tenenbaum *et al.*, 2000), Laplacian Eigenmaps (Belkin and Niyogi, 2003), local linear embedding (Saul and Roweis, 2003, LLE), local tangent space alignment (Zhang and Zha, 2004, LTSA), and Diffusion Map (Nadler *et al.*, 2006). All of these methods aim to find a low-dimensional representation of the given data, but they utilize different strategies towards achieving it. Isomap finds a lower-dimensional embedding that best preserves the geodesic distance between all points, while Laplacian Eigenmaps tries to preserve local distances. LLE seeks to maintain neighborhood distances. LTSA is algorithmically similar to LLE but tries to learn local neighborhood geometry via tangent spaces and aligns them to find the underlying manifold. Diffusion map uses a different perspective by considering a random walk "diffusing" through the points, and uses a particular

eigendecomposition related to that walk to obtain the low dimensional embedding. Spanifold (Chenouri *et al.*, 2015) sets up a tree on the manifold and tries to maintain pairwise distance relationships within the tree while flattening the manifold. In Section 2.3.1 we will compare the performance of these different approaches on the ADAPT data.

Once the domain, $\mathbf{M}_0$, is obtained, the next step is to construct $\mathbf{Y}_n : \mathbf{M}_0 \to \mathbb{R}^D$ from $\mathcal{Y}_n$. As functional data are commonly expressed with basis functions, we fix basis functions $\mathbf{e}_j : \mathbf{M}_0 \to \mathbb{R}^D$ and then we express the manifolds in functional data format as

$$\mathcal{Y}_n \equiv \mathbf{Y}_n(m) \approx \sum_{j=1}^{J} b_{nj}\mathbf{e}_j(m) \qquad m \in \mathbf{M}_0,$$

where $b_{nj} \in \mathbb{R}$. The number of basis functions J should be high enough so that the original details are preserved. The $\hat{b}_{nj}$ can be found by minimizing

$$\sum_{p=1}^{P} |\mathrm{y}_{np} - \mathbf{Y}_n^J(m_p)|^2 + \lambda \int_{\mathbf{M}_0} [L(\mathbf{Y}_n^J)(m)]^2 dm \tag{2.1.1}$$

where $\mathrm{y}_{np} = (\tilde{\mathrm{y}}_{np1}, \cdots, \tilde{\mathrm{y}}_{npD})^\top$, $\mathbf{Y}_n^J(m) = \sum_{j=1}^{J} b_{nj}\mathbf{e}_j(m)$, λ is a smoothing parameter, and L is a linear differential operator. The resulting functional data would be

$$\mathbf{Y}_n(m) \approx \sum_{j=1}^{J} \hat{b}_{nj}\mathbf{e}_j(m). \tag{2.1.2}$$

In the ADAPT application we utilize felsplines (Ramsay, 2002) and expand each coordinate, $Y_{nj}(m)$, separately, though other approaches including thin plate splines could also be used.

2.1.2 2-Step Functional Principal Component Analysis

We now introduce a 2-step functional principal component analysis (FPCA) method to be carried out on the objects define in (2.1.2). In the first step, we conduct FPCA on the pooled (across the D coordinates) sample to reduce the number of basis functions and make them orthogonal. In the second step, we conduct PCA on the resulting array to get eigenvalues λ_k and eigenfunctions $\mathbf{V}_k(m)$. Computational tools for basis functions that map a set in lower dimension $\mathbf{M}_0 \in \mathbb{R}^d$ to higher dimension, $\mathbb{R}^D$, are currently limited. Therefore, we start with expanding each coordinate of $\mathbf{Y}_n$ using basis functions $\{e_j : \mathbf{M}_0 \to \mathbb{R}\}$, and then obtain eigenfunctions $\mathbf{V}_k : \mathbf{M}_0 \to \mathbb{R}^D$ in the second step.

The raw data, $\{\mathrm{y}_{npq}\}$, is assumed to be an $N \times P \times D$ array, while the basis coefficients from (2.1.2) form an $N \times J \times D$ array. Our second step consists of tensor multiplication and singular value decompositions, which are substantial computational burdens. Decreasing the dimension by lowering the number of basis functions through the first step lessens the computational time substantially. The burden of the second step is also decreased substantially by exploiting the orthonormal structure of the bases from the first step.

Step 1. Without loss of generality, assume that the $\mathbf{Y}_n(m)$ have been centered and thus have mean zero. Each of the functions is expressed as

$$\mathbf{Y}_n(m) = [Y_{n1}(m), \cdots, Y_{nD}(m)]^\top \approx \left[\sum_{j=1}^{J} \hat{b}_{nj1} e_j(m), \cdots, \sum_{j=1}^{J} \hat{b}_{njD} e_j(m)\right]^\top,$$

for $n = 1, \ldots, N$. We stack all of the coordinate-wise functions into a single vector of functions with dimension ND. We denote the resulting functions as $Y_l(m)$ where $Y_l(m) = Y_{nq}(m)$ for $l = N(q-1) + n$ and $n = 1, \cdots, N$, $q = 1, \cdots, D$.

We find the pairs of eigenvalues η_h and principal component functions $\psi_h : \mathbf{M}_0 \to \mathbb{R}^D$, for $h = 1, \ldots, H$, which satisfy

$$\eta_h \psi_h(m) = \int \mathbf{\Phi}(m, m')\psi_h(m')dm', \tag{2.1.3}$$

$$\|\psi_h\| = 1, \tag{2.1.4}$$

where $\mathbf{\Phi}(m, m') = E[Y_l(m)Y_l(m')^\top] = \sum_{j_1}^J \sum_{j_2}^J E[b_{lj_1} b_{lj_2}] e_{j_1}(m) e_{j_2}(m')^\top$.

We now can expand $Y_{nq}(m)$ using the $\{\psi_h(m)\}$:

$$Y_{nq}(m) = \sum_{h=1}^{H} c_{nhq}\psi_h(m) \tag{2.1.5}$$

where H is much less than J and $\psi_h(m)$'s are orthonormal while $e_j(m)$ are not. This purely serves to reduce the computational burden of the second step.

Step 2. From the representation (2.1.5), the coefficients $\mathbf{c} = \{c_{nhq}\}$ form an $N \times H \times D$ array. The covariance operator of $\mathbf{Y}_n(m)$ is given by

$$\Gamma_{q,q'}(m, m') = E[Y_{nq}(m)Y_{nq'}(m')] = \sum_{h_1=1}^{H} \sum_{h_2=1}^{H} \Sigma_{h_1 q h_2 q'} \psi_{h_1}(m)\psi_{h_2}(m')$$

with $\Sigma_{h_1 q h_2 q'} = E[c_{nh_1 q} c_{nh_2 q'}^\top]$, a $H \times D \times H \times D$ tensor. Now we find the pairs of eigenvalues λ_k and eigenfunctions $\mathbf{V}_k(m)$ that satisfy

$$\lambda_k \mathbf{V}_k(m) = \int \mathbf{\Gamma}(m, m')\mathbf{V}_k(m')dm', \tag{2.1.6}$$

$$\|\mathbf{V}_k\| = 1. \tag{2.1.7}$$

Details of the algorithm are enclosed in Section 2.4.1.

2.1.3 Manifold-on-Scalar Regression

We now give a Manifold-on-scalar regression strategy by using the functional manifold objects as responses and using scalar predictors, very similar to Function-on-Scalar Regression (Ramsay and Silverman, 2005). The model is given by

$$\mathbf{Y}_n(m) = x_{n1}\boldsymbol{\beta}_1(m) + x_{n2}\boldsymbol{\beta}_2(m) + \cdots + x_{nR}\boldsymbol{\beta}_R(m) + \boldsymbol{\varepsilon}_n(m), \tag{2.1.8}$$

where there are R predictors for every manifold. Recall that $\mathbf{Y}_n$ is the deformation map associated with the manifold, $\mathcal{Y}_n$.

While least squares works well for finding $\hat{\beta}$'s, it can often be improved by penalizing the roughness of the resulting $\hat{\beta}$'s. In this case the objective function will be

$$\sum_{n=1}^{N}\int_{\mathbf{M}_0}\left|\mathbf{Y}_n(m) - \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_r(m)\right|^2 dm + \sum_{r=1}^{R}\lambda_r\int_{\mathbf{M}_0}|L\boldsymbol{\beta}_r(m)|^2 dm, \tag{2.1.9}$$

where λ_r is tuning parameter and L is a roughness operator. It is important to choose L carefully. Since we do not want the minimizer of (2.1.9) to change based on the coordinate system, we need an operator that is invariant to rotation and translation. For this reason Ramsay (2002) chose the Laplacian operator. In the case where $\mathbf{f} : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}^3$, the Laplacian operator of $\mathbf{f}$ is given by

$$\begin{aligned}\triangle\mathbf{f} &= \triangle[f_1, f_2, f_3]^\top = [\triangle f_1, \triangle f_2, \triangle f_3]^\top \\ &= \left[\frac{d^2 f_1}{dm_1^2} + \frac{d^2 f_1}{dm_2^2}, \frac{d^2 f_2}{dm_1^2} + \frac{d^2 f_2}{dm_2^2}, \frac{d^2 f_3}{dm_1^2} + \frac{d^2 f_3}{dm_2^2}\right]^\top,\end{aligned}$$

where f_1, f_2, f_3 correspond to each coordinate of $\mathbf{f}$ and m_1 and m_2 correspond to each coordinate of $\mathbf{M}_0$. $\mathbf{Y}_n$ and β_r can both be expanded with PC functions

$\mathbf{V}_k : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}^3$ for $k = 1, \cdots, K$:

$$\sum_{n=1}^{N} \int_{\mathbf{M}_0} \left| \sum_{k=1}^{K} y_{nk} \mathbf{V}_k(m) - \sum_{r=1}^{R} x_{nr} \sum_{k=1}^{K} b_{rk} \mathbf{V}_k(m) \right|^2 dm + \sum_{r=1}^{R} \lambda_r \int_{\mathbf{M}_0} \left| \sum_{k=1}^{K} b_{rk} (\triangle \mathbf{V}_k(m)) \right|^2 dm. \tag{2.1.10}$$

Then objective (2.1.10) becomes to find B that minimizes

$$trace\{(\mathbf{y} - \mathbf{X}B)^\top (\mathbf{y} - \mathbf{X}B)\} + trace\{\Lambda B U B^\top\},$$

where $\mathbf{y}$ is $N \times K$ matrix of y_{nk}'s, $\mathbf{X}$ is $N \times R$ matrix of x_{nr}'s, B is $R \times K$ matrix of b_{rk}'s, Λ is a diagonal matrix of λ_r's, and $U_{k_1,k_2} = \int_{\mathbf{M}_0} (\triangle \mathbf{V}_{k_1})^\top (\triangle \mathbf{V}_{k_2}) dm$. Then the least square estimate of B is

$$\mathrm{vec}(\hat{B}^\top) = \left((\mathbf{X}^\top \mathbf{X}) \otimes I_K + \Lambda \otimes U^\top \right)^{-1} \mathrm{vec}(\mathbf{y}^\top \mathbf{X}).$$

See Section 2.4.2 for further details.

Now we consider testing for $\hat{\beta}$'s. Here we present two testing methods: pointwise significance and overall significance. This will later be illustrated by plots in Section 2.3.3. To test for the significance of $\hat{\boldsymbol{\beta}}$, we find the asymptotic distribution of $\hat{\boldsymbol{\beta}}$. Assume that

$$Y_n(m) = \mathbf{X}_n^\top \boldsymbol{\beta}(m) + \epsilon_n(m),$$

where $\{\mathbf{X}_n\}$ are iid random elements of $\mathbb{R}^R$ whose covariance matrix, $\Sigma_{\mathbf{X}}$, exists and has full rank. Also assume that $\{\epsilon_n\}$ are mean zero iid elements of $L^2[\mathbf{M}_0]$ with $E\|\epsilon_n\|^2 < \infty$, which implies the covariance function, $\mathbf{C}_\epsilon(m, m')$, of $\epsilon_n(m)$ exists.

Assume that the sequences $\{\mathbf{X}_n\}$ and $\{\epsilon_n\}$ are independent of each other. Then we have by the CLT for Hilbert spaces that

$$\sqrt{N}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(0, \mathbf{C}_\beta),$$

where $\mathbf{C}_\beta(m, m')$ is an $R \times D \times R \times D$ array at each pair $(m, m') \in \mathcal{M}_0 \times \mathcal{M}_0$ and equals

$$C_{i,j,k,l;\beta}(m, m') = (\Sigma^{-1})_{i,k;\mathbf{X}} C_{j,l;\epsilon}(m, m').$$

The covariance of the errors can be estimated as

$$\hat{\mathbf{C}}_\epsilon(m, m') = \frac{1}{N - R} \sum_{n=1}^{N} (\mathbf{Y}_n(m) - \mathbf{X}_n^\top \hat{\boldsymbol{\beta}}(m))(\mathbf{Y}_n(m') - \mathbf{X}_n^\top \hat{\boldsymbol{\beta}}(m'))^\top.$$

Notice this $\hat{\mathbf{C}}_\epsilon$ corresponds with a $P \times D \times P \times D$ array because at every $(m_p, m_{p'})$, $\hat{\mathbf{C}}_\epsilon(m_p, m_{p'})$ is $D \times D$ matrix, and there are P points m_p. For each $\beta_r(m)$,

$$\sqrt{N}(\hat{\beta}_r - \beta_r) \xrightarrow{d} \mathcal{N}(0, \mathbf{C}_\beta^r),$$

and we estimate

$$\hat{\mathbf{C}}_\beta^r(m, m') = N(\mathbf{X}^\top \mathbf{X})_{r,r}^{-1} \hat{\mathbf{C}}_\epsilon(m, m').$$

We now test the significance of $\hat{\beta}_r$. We can do this pointwise, i.e. test $\hat{\beta}_r(m_p)$ at each point, and we can also find the overall confidence region around the face using the strategy of Choi and Reimherr (2018). We call this simultaneous confidence region a *confidence bubble* as it forms a 3D region around the parameter estimates. We first rotate $\hat{\beta}_r$ and get $\hat{\beta}_r' = (\hat{\mathbf{C}}_\beta^r(m, m))^{-1/2} \hat{\beta}_r$. Then

$$\sqrt{N}(\hat{\beta}_r' - \beta_r') \xrightarrow{d} \mathcal{N}(0, \tilde{\mathbf{C}}_\beta^r)$$

where

$$\tilde{\mathbf{C}}^r_\beta(m,m') = (\hat{\mathbf{C}}^r_\beta(m,m))^{-1/2}\hat{\mathbf{C}}^r_\beta(m,m')(\hat{\mathbf{C}}^r_\beta(m',m'))^{-1/2}.$$

Pointwise we have

$$\sqrt{N}(\hat{\beta}'_r(m_p) - \beta'_r(m_p)) \xrightarrow{d} \mathcal{N}(0, I_{3\times 3}).$$

Therefore,

$$T^{pt}_{norm} = N\|(\hat{\beta}'_r(m_p) - \beta'_r(m_p)\|^2 \xrightarrow{d} \chi^2(3).$$

We conclude by providing a strategy for constructing simultaneous confidence ellipses for each $\hat{\beta}(m)$, which is based on a technique from Choi and Reimherr (2018). We call a testing based on this as Choi test. The proof can be found in Section 2.4.3.

Theorem 2.1.1. *If $\sqrt{N}(\hat{\beta}_r - \beta_r)$ converges in distribution to a Gaussian process, $\mathcal{N}(0, \mathbf{C}^r_\beta)$, and the square-root of the eigenvalues, $\{\lambda_i\}$, of $\mathbf{C}^r_\beta$ are summable, then*

$$P\left\{\sqrt{N}|\hat{\beta}_r(m) - \beta_r(m)| \leq \sqrt{\xi_\alpha \sum_{j=1}^{\infty} \sqrt{\lambda_j}|\mathbf{U}_j(m)|^2}, \text{ for almost all } m \in \mathbf{M}_0\right\} \leq \alpha + o(1),$$

where $\{\mathbf{U}_j\}$ are the eigenfunctions of $\mathbf{C}^r_\beta$, and ξ_α is such that $P(\sum_{j=1}^{\infty}\sqrt{\lambda_j}Z_j^2 > \xi_\alpha) \xrightarrow{d} \alpha$.

2.2 Simulation Studies

In this Section, we compare the performance of manifold-on-scalar regression to the performance of multivariate principal component regression, PCR, to show that the functional approach works better than the multivariate approach. Our only comparison is done with the multivariate PCR since no method is found such

that takes a whole shape or manifold as a variable for regression. Most previous methods extract some features from face and take them as variables or focus on much localized parts of face.

We construct simulated faces using the following model:

$$Y_n(m_{ni}) = \delta \times X \times \beta(m_{ni}) + \epsilon(m_{ni}) + \gamma_{ni}.$$

Here δ is a positive constant signifying the strength of the effect, $X \sim \mathcal{N}(0,1)$, $\beta(m)$ is a coefficient function, $\epsilon(m)$ is an error function, and γ_{ni} is an iid measurement noise. To ensure a realistic simulation, we take $\beta(m)$ to be the estimated $\beta(m)$ for height from Section 2.3. A plot of this β is given in the top left of Figure 2.3. The error, $\varepsilon(m)$ is constructed by randomly selecting one of the faces from the ADAPT study, while the γ_{ni} is a vector of $\mathcal{N}(0, 0.002)$ displacements to each x, y, and z coordinate. The examples of simulated faces are as in Figure 2.2. The plots shown are based on $\delta = 20$.

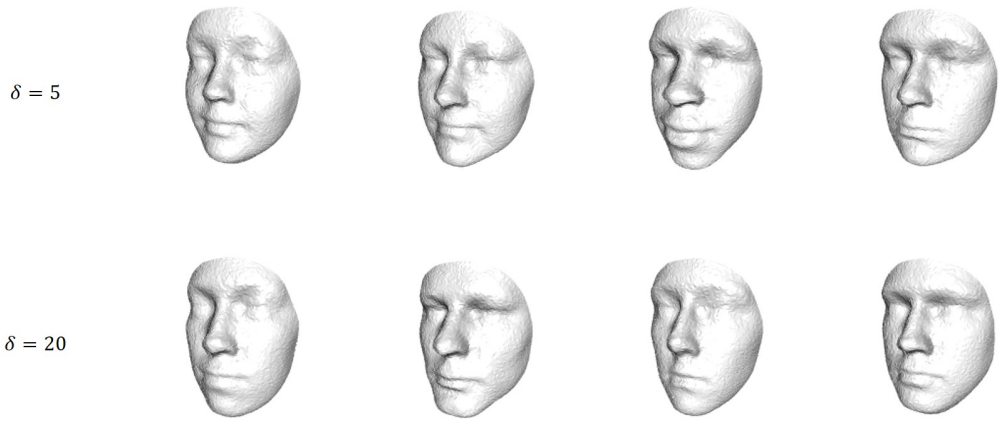

Figure 2.2: Examples of simulated faces for $\delta = 5$ (top) and $\delta = 20$ (bottom).

We repeated the simulation 1000 times for $\delta = 0$, $\delta = 5$, $\delta = 20$, $\delta = 50$,

$\delta = 100$, and $\delta = 200$ and for each run $N = 100$ is taken. We compared the rejection rates based on PCA test, Choi test, and norm test for multivariate PCR and functional PCR. For multivariate PCR, we ran principal component analysis on pooled data, stacking x-coordinate values, y-coordinate values, and z-coordinate values of the 7150 quasi-landmarks on the faces. We then took principal components that explain 99 % of the total variation, used them as our response, and fit a linear regression model with predictor X. We have compared this to our method, functional PCR. It is not very conventional to conduct a PCA test, Choi test, or norm test in the multivariate case, since those tests target infinite dimensional spaces. However, to make a proper comparison, we have used the same testing methods.

The results are summarized in Table 2.1. It shows that the rejection rates for $\delta = 0$ is within 2 standard error of 0.05, the alpha level we took. As δ increases, the rejection rate increases as expected, and when $\delta = 200$ the rejection rate becomes almost 1. In most of cases, the rejection rates for functional PCR are bigger than the rejection rates for multivariate PCR, except a few cases like Choi test for $\delta = 20$.

Some examples of estimated betas from functional PCR and from multivariate PCR are given in Figure 2.3. Red and yellow colors are where beta shows outward effect, meaning that in those parts the face goes outward when predictor increases, while blue and skyblue means that beta shows inward effect. Since it is the coefficient function for height that we have used, the plot in left shows that the face would become prolonged as the predictor increases. And the plots show that the estimated beta from functional PCR picks up the smoothness of the original beta and better resembles the original beta, while the estimated beta from multivariate PCR shows rough edges and sometimes gives very different effect as in the bottom

δ	type	rej rate PCA	rej rate Choi	rej rate norm
0	multivartate	0.044	0.061	0.038
	functional	0.059	0.064	0.060
5	multivariate	0.054	0.070	0.049
	functional	0.078	0.078	0.069
20	multivariate	0.063	0.090	0.061
	functional	0.097	0.101	0.094
50	multivariate	0.136	0.221	0.181
	functional	0.265	0.299	0.293
100	multivariate	0.474	0.739	0.596
	functional	0.662	0.768	0.745
200	multivariate	0.989	1.000	0.997
	functional	0.999	1.000	0.999

Table 2.1: The rejection rates based on three different tests (PCA test, Choi test, and norm test) for different δ's. For $\delta = 0$ case, the rejection rates are approximately 0.05, the alpha in this case, and for the other cases, the rejection rates for functional PCR are higher than the rejection rates for multivariate PCR.

($\delta = 20$ case).

2.3 ADAPT Study

This Section presents the application of our methodologies from Section 2.1 to the ADAPT data. We convert the 3D facial imaging data into functional objects in Section 2.3.1, where we also discuss the details on how to apply each step of the framework in Section 2.1.1. Section 2.3.2 presents the principal components of our 2-step FPCA from Section 2.1.2. Section 2.3.3 presents a regression model with the 3D faces as manifold outcomes and the covariates age, gender, height, weight, and genetic ancestry; we discuss the effects and significances of the resulting coefficient functions.

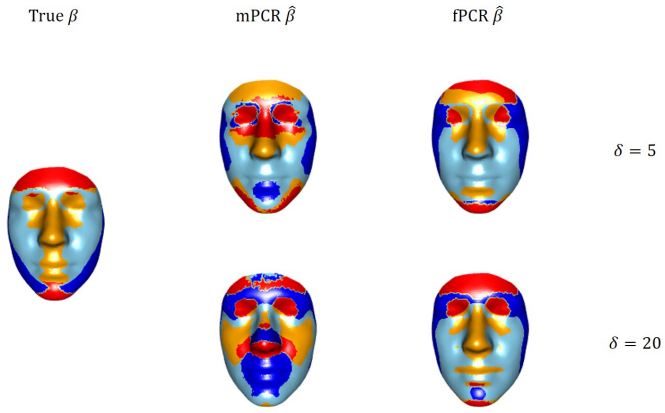

Figure 2.3: Examples of estimated beta. The left is the beta used for simulation, the middle is the estimated beta from multivariate PCR, and the right is the estimated beta from functional PCR. The top row is for $\delta = 5$ and the bottom row is for $\delta = 20$. Red and yellow means that beta shows outward effect while blue and skyblue means that beta shows inward effect.

2.3.1 Facial Functional Object Construction

We view each face as a 2-dimensional manifold that is a subset of $\mathbb{R}^3$, and our goal is to construct functional objects $\mathbf{Y}_n : \mathbf{M}_0 \to \mathbb{R}^3$ from each face. There are 6564 faces, and each face is sampled densely with 7150 points in x, y, and z coordinates. Therefore, the data is $\{\mathrm{y}_{npq} : n = 1, \cdots, 6564; p = 1, \cdots, 7150; q = 1, 2, 3\}$. We elaborate each step of constructing facial functional objects as below.

Step 1. We identified a reference face $\mathcal{M}_0$ as the mean of the 6564 faces, that is, $\{\bar{\mathrm{y}}_{pq} : \bar{\mathrm{y}}_{pq} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{y}_{npq}; p = 1, \cdots, 7150; q = 1, 2, 3\}$. This approach is possible because the data were already aligned via Procrustes analysis; the resulting mean face is given in Figure 2.4.

Step 2. We apply manifold learning techniques to the mean face to find $\mathbf{M}_0$, the representation of mean face in $\mathbb{R}^2$. The resulting $\mathbf{M}_0$ is represented by

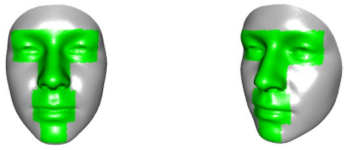

Figure 2.4: Plots of mean face which is taken as a reference face. Green area represents the area where finer mesh is taken for felspline basis functions. Examples of mesh plots are as in Figure 2.5.

$\{\mathrm{m}_{pq}; p = 1, \cdots, 7150, q = 1, 2, 3\}$. The choice of the manifold learning technique for constructing $\mathbf{M}_0$ is important to obtain a reasonable functional object that is close to the original data. Figure 2.5 shows how $\mathbf{M}_0$ changes with different manifold learning techniques. Since smoothness is defined with distance along $\mathcal{M}_0$, we believed that the manifold learning techniques that preserve local distances would work best. In order to check our intuition, we tried several nonlinear dimension reduction techniques like local linear embedding (LLE, Saul and Roweis (2003)), Laplacian eigenmaps (Belkin and Niyogi, 2003), Isomap (Tenenbaum *et al.*, 2000), local tangent space alignment (LTSA, Zhang and Zha (2004)), Diffusion Map (Nadler *et al.*, 2006), and Spanifold Chenouri *et al.* (2015) along with a linear dimension reduction technique principal component analysis (PCA) for a comparison.

Step 3. We construct basis functions $\mathbf{e}_j : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}^3$, but given the limitations in constructing such basis functions, we took basis functions

$e_j : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}$ to expand the functional objects marginally

$$\mathbf{Y}_n(m) = \begin{bmatrix} Y_{n1} \\ Y_{n2} \\ Y_{n3} \end{bmatrix} (m) = \sum_{j=1}^{J} \begin{bmatrix} b_{nj1} \\ b_{nj2} \\ b_{nj3} \end{bmatrix} e_j(m) \tag{2.3.1}$$

where Y_{n1} corresponds to x coordinate of $\mathbf{Y}_n$, Y_{n2} corresponds to y coordinate of $\mathbf{Y}_n$, and Y_{n3} corresponds to z coordinate of $\mathbf{Y}_n$. We find $\{\hat{b}_{njq}\}$ for $q = 1, 2, 3$ by minimizing (2.1.1). We used felsplines (Ramsay, 2002) which are designed for irregularly shaped domain with complex boundaries and use a finite element method, meaning that the domain is divided into triangular meshes and piecewise linear and quadratic functions are fit on each mesh. Therefore, we needed to create meshes out of our domain. Ramsay (2002) uses all data points as vertices of the mesh, but in our case that will return over twenty thousand basis functions. In order to limit the number of basis functions to less than the number of observations per face, which is 7150 in our data, we created new meshes using the `R` package `INLA` (Lindgren and Rue, 2013).

There can be many different ways to create meshes, and the choice of mesh is closely related to the number of basis functions, thus affecting how close the functional objects are to the data. We took a finer mesh around periorbital, perinasal, and perioral areas shown as the green area in Figure 2.4, as these localized facial features are emphasized in (Hammond *et al.*, 2005), and a coarser mesh around the cheeks and forehead where the surface is more smooth. The meshes for different manifold learning techniques are given in the bottom row of plots of Figure 2.5.

Table 2.2 presents the average mean squared errors (AMSE) of 100 randomly

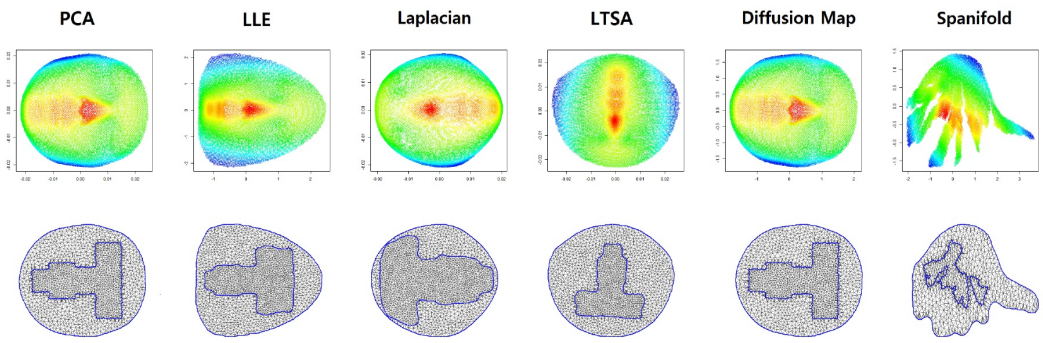

Figure 2.5: The dimension-reduced reference manifold $\mathbf{M}_0$ and corresponding mesh using different manifold learning techniques. The finer inner mesh correspond to the area of green in Figure 2.4.

selected faces, which is a measure of how close the functional objects $\mathbf{Y}_n(m)$ are to the original data $\{\mathrm{y}_{npq}\}$:

$$\text{AMSE:} \qquad \frac{1}{N}\frac{1}{P}\sum_{n=1}^{N}\sum_{p=1}^{P}|\mathrm{y}_{np}-\mathbf{Y}_n(m_p)|^2, \tag{2.3.2}$$

where $N = 100$ and $P = 7150$. We stress that, at this stage, we are not aiming for dimension reduction; our goal is to approximate the data using basis functions with as little error as possible. Therefore, in this step we want the AMSE to be as small as possible to minimize any information loss when converting to functional objects. However, the Procustes Analysis used to initially align and scale the data results in a unit-less scale for the coordinates of the face; the x-axis has been rescaled to have a range of 1. This means that the AMSE values themselves are difficult to interpret, and thus we focus on comparisons of the AMSE's. The range of the first coordinate of the domain points $\mathbf{m}_n$, $\{\mathrm{m}_{p1}\}$, is different for each $\mathbf{M}_0$ from the different manifold learning techniques, and thus we made the smoothing parameter, λ, in (2.1.1) dependent on the range of x. The AMSE for LTSA with λ_3 is smallest, while the AMSE for LLE with λ_1 is also similarly small. Both LTSA

and LLE try to preserve neighborhood distances of the original manifold, which confirms our intuition that they would best represent smoothness defined with distances along $\mathbf{M}_0$ and thus give a good fit. Spanifold gives the largest AMSE, which is not surprising given that the $\mathbf{M}_0$ is very irregular. This is because a human face has many local peaks, and Spanifold works better with more regular surfaces.

	PCA	LLE	Laplacian	LTSA	Diffusion Map	Spanifold
λ_1	0.00247	0.00018	0.00423	0.00078	0.00056	0.00679
λ_2	0.00084	0.00036	0.00167	0.00017	0.00078	0.00690
λ_3	0.00058	0.00083	0.00114	**0.00013**	0.00142	0.00772
range(x)	0.046	3.900	0.041	0.051	3.906	5.671

Table 2.2: The pointwise mean squared errors of $\mathbf{Y}(m)$ of 100 randomly selected faces as in (2.3.2) for different λ's from mesh as in Figure 2.5. $\lambda_1 = \text{range(x)}/10^4$, $\lambda_2 = \text{range(x)}/10^5$, $\lambda_3 = \text{range(x)}/10^6$ where range(x) is the range of $\{\text{m}_{p1}\}$.

For all subsequent analyses, we utilize the manifold objects constructed using the presented LTSA mesh and used λ a little less than λ_3 to recover the details of face. Figure 2.7 shows that the facial functional objects are very close to the original faces except for some smoothing. Figure 2.6 shows a heatmap of the pointwise errors between functional objects and the original data. The tip of the nose shows a relatively high pointwise error compared to the other areas, which is due to smoothing. The boundary does not show much deviation and seems to be stable. We believe the resulting objects are reasonable approximations of the original faces.

2.3.2 Functional Principal Component Analysis

In this Section we apply the 2-step Functional Principal Component Analysis (2-step FPCA) discussed in Section 2.1.2 to the ADAPT data. We take $H = 200$

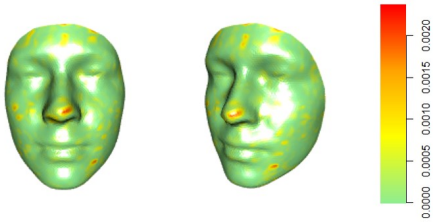

Figure 2.6: Plot shows pointwise mean squared errors across all 6564 faces. This shows that the difference between the original faces and facial functional objects are very small.

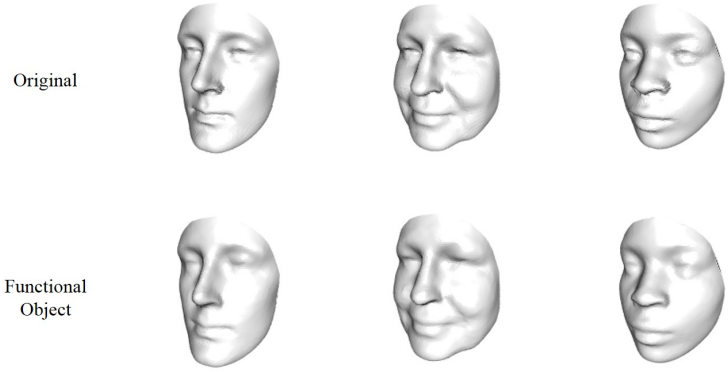

Figure 2.7: Top three plots are examples of facial data of ADAPT, $\{y_{npq}\}$, and bottom three plots are corresponding facial functional objects, $\mathbf{Y}_n(m)$. This shows that the facial functional objects closely resembles the original faces.

principal components, or $\psi_h(m)$, in the first step (pooling coordinates), which accounts for 99.9% of the total variance. In the second step we then compute the PCs without pooling coordinates,$\mathbf{V}_k(m)$, and Figure 2.8 shows the cumulative proportion of explained variance. The first principal component $\mathbf{V}_1(m)$ explains 31.27%, the second principal component $\mathbf{V}_2(m)$ explains 12.43%, and the third principal component $\mathbf{V}_3(m)$ explains 10.59% of variation. The first 5 principal components combined explain 66.71%, and the first 10 principal components combined explain 81.26%.

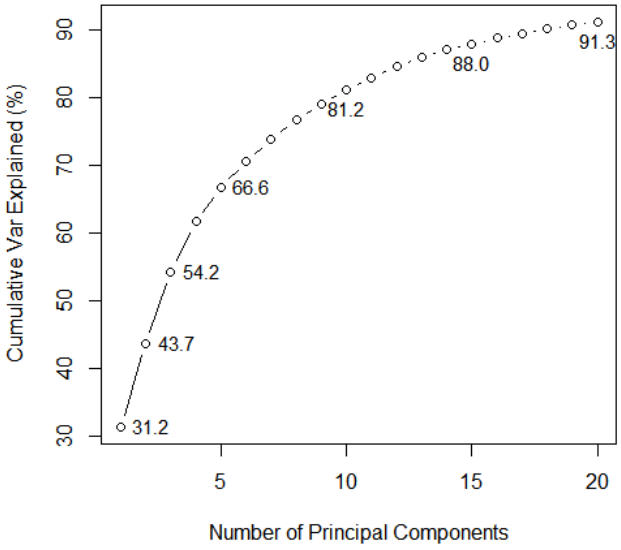

Figure 2.8: Cumulative proportion of variance for number of PCs. First 10 PCs explain about 81.2% of total variance and first 18 PCs explain about 90.2% of total variance.

In Figure 2.9, we demonstrate how each principal component affects the face, which is a bit challenging to visualize given that we are working in 3D. We thus compute the orthogonal vector, $\mathbf{t}_p$, to the tangent plane of each facial point, m_p, by conducting traditional PCA in a small neighborhood of m_p (distance 0.1). As the first and second principal components would be the two vectors spanning the tangent plane, the third principal component would be the vector orthogonal to the tangent plane. Note that PCA also gives $|\mathbf{t}_p|^2 = 1$. We then calculated the inner product $\langle \mathbf{V}_k(m_p), \mathbf{t}_p \rangle$ at each point for $p = 1, \cdots, 7150$. The yellow to red area in Figure 2.9 denotes a PC whose effect points outward while the lightblue to blue area means that the effect of PC at that point is inward. Orange and lightblue mean weaker effects and red and blue mean stronger effects.

As the top leftmost plot of Figure 2.9 suggests, the major difference between the

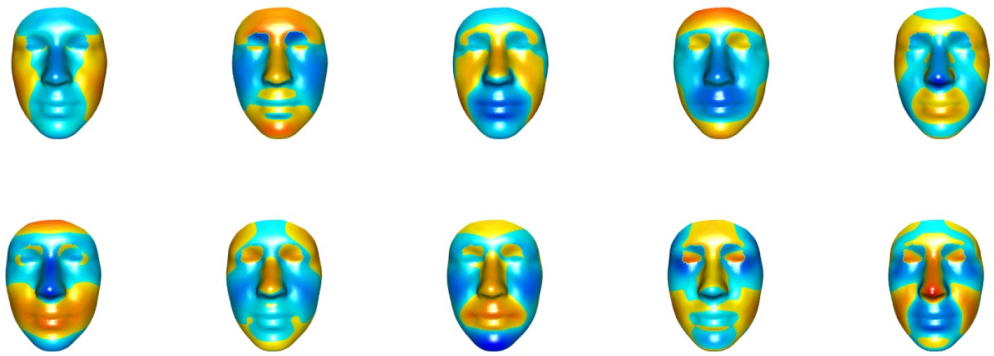

Figure 2.9: The directional plots for PC 1-5 on the top and PC 6-10 on the bottom. The color on the face shows the direction and the strength, from weakest to strongest, of each PC effect on face: from lightblue to blue, inward, and from yellow to red, outward.

mean face and the reconstructed faces using the first PC in Figure 2.10 is the sides of the faces. The top face became thinner while the middle face became a little rounder on the cheek. Figure 2.10 shows the progression of facial changes with more PCs included. The rightmost faces are good approximations to the bottom plots in Figure 2.7, explaining 91.39% of total variation. Thus we have now reduced the dimension of the data from 7150 points to 20 principal components, while carefully controlling the information loss.

2.3.3 Manifold-on-Scalar Regression

We conclude the application Section by carrying out Manifold-on-Scalar Regression, which represents a major strength of our methodology. We examine the effects of sex, age, height, weight, and genetic ancestry the structure of human faces. Genetic ancestry is measured as a proportion of a particular ethnic background, where E.ASN refers to East Asian, S.ASN refers to South Asian, AMR refers to Native American, W.AFR refers to West African, and S.EUR refers Southern European.

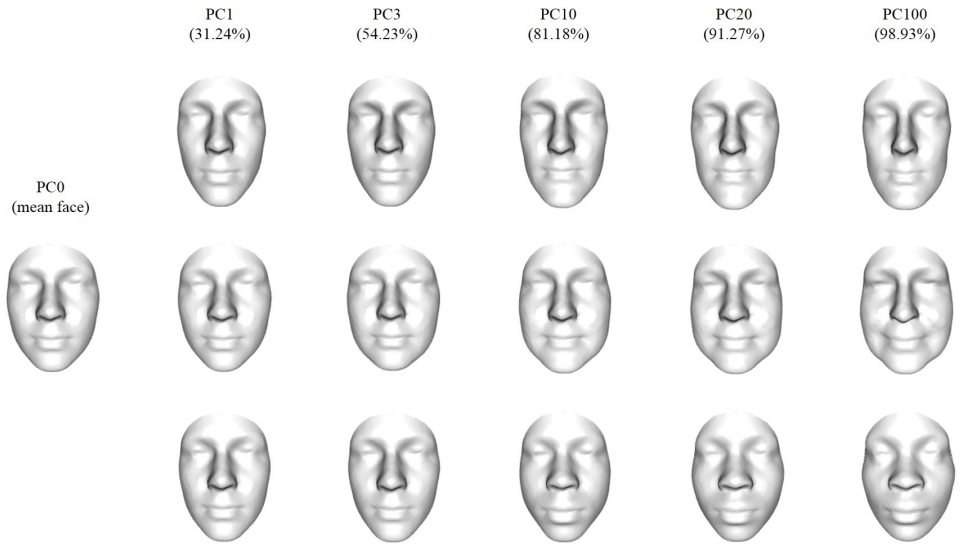

Figure 2.10: Three facial functional objects expanded using different number of PCs from the second step of FPCA. Leftmost plot is the mean face. The percentage of variation explained is given at the top of each column.

There is also N.EUR which refers to Northern European, but since the sum of all proportions is 1, it is removed from the covariates, meaning that it is acting as the ancestral baseline, so all ancestral effects indicate differences from Northern Europeans. For the response variable, we take the facial functional objects $\mathbf{Y}_n(m)$ expanded with $K = 100$ principal components from the FPCA in Section 2.3.2. The model is as in (2.3.3). Since the genetic ancestry was not computed for all individuals, the number of facial manifolds involved in the model is $N = 3287$. The model also includes a $\mathcal{N}(0, 10)$ noise variable just as a check to make sure our subsequent p-values have proper specificity.

$$\begin{aligned}\mathbf{Y}_n(m) &= \beta_0(m) + \beta_1(m)\text{sex}_n + \beta_2(m)\text{age}_n + \beta_3(m)\text{height}_n + \beta_4(m)\text{weight}_n \\ &+ \beta_5(m)p_n^{\text{E.ASN}} + \beta_6(m)p_n^{\text{S.ASN}} + \beta_7(m)p_n^{\text{AMR}} + \beta_8(m)p_n^{\text{W.AFR}} + \beta_9(m)p_n^{\text{S.EUR}} \\ &+ \beta_{10}(m)(\text{sex}_n \times \text{age}_n) + \beta_{11}(m)(\text{age}_n \times \text{weight}_n) \\ &+ \beta_{12}(m)(\text{height}_n \times \text{weight}_n) + \beta_{13}(m)\text{noise}_n + \epsilon_n(m).\end{aligned} \tag{2.3.3}$$

We estimated beta functions β_r's with regularization term as outlined in Section 2.1.3. The tuning parameter λ_r's are determined using iterative 4-fold cross validation.

The sizes and p-values of resulting $\hat{\beta}_r$ are presented in Table 2.3. We utilize three different tests as outlined in Choi and Reimherr (2018). Each test uses slightly different normalizations of the estimated parameter functions. The first test is based on the L^2-norm, which ignores the covariance operator of the parameter estimate (though it is used in calculating p-values). The other two approaches attempt to normalize by the covariance operator, where the PC and Choi approach normalize by the Moore-Penrose inverse of the covariance operator and square-root of the covariance operator, respectively. Both approaches can be phrased using PCA, and we refer the interested reader to Choi and Reimherr (2018) for more details.

The asymptotic distribution of the PC approach is simply a chi-squared distribution, while the norm and Choi approach are given by weighted sums of chi-squares. We approximate p-values from the weighted distribution using Imhof's method (Imhof, 1961; Duchesne and Lafaye de Micheaux, 2010). The p-values suggest that all beta functions are significant at 99% significance level except the noise. Therefore, the tests seem to have discerned the effects from the true negative

variable.

	Predictor	$\|\hat{\beta}_r\|^2$	p-value (PC)	p-value (Choi)	p-value (Norm)
β_0		3.910e-05	$<$ 1.110e-21	5.385e-15	2.631e-09
β_1	sex	9.924e-07	$<$ 1.110e-21	5.551e-16	1.443e-15
β_2	age	4.443e-10	5.888e-11	3.053e-15	3.672e-04
β_3	height	2.105e-09	$<$ 1.110e-21	5.551e-17	3.514e-14
β_4	weight	1.549e-09	9.826e-08	6.088e-06	1.173e-02
β_5	$p^{\text{E.ASN}}$	2.439e-06	$<$ 1.110e-21	4.996e-15	3.164e-15
β_6	$p^{\text{S.ASN}}$	5.859e-07	1.332e-17	2.220e-16	1.720e-11
β_7	p^{AMR}	7.689e-07	1.418e-21	1.110e-16	1.357e-08
β_8	$p^{\text{W.AFR}}$	1.783e-06	$<$ 1.110e-21	1.110e-15	7.772e-16
β_9	$p^{\text{S.EUR}}$	1.514e-06	$<$ 1.110e-21	6.661e-16	8.826e-14
β_{10}	sex $\times$ age	2.103e-10	1.988e-19	5.551e-17	2.034e-10
β_{11}	age $\times$ weight	3.523e-14	7.790e-06	5.328e-11	9.143e-03
β_{12}	height $\times$ weight	7.553e-14	3.521e-10	5.074e-07	1.488e-03
β_{13}	noise	4.188e-12	3.419e-01	3.242e-01	5.647e-01

Table 2.3: The size of $\hat{\beta}_r$ and p-values based on PC test, Choi test, and Norm test are presented.

Now that we have carried out our hypothesis testing, it is important to visualize and more fully understand the estimated beta functions. Since these functions have domain of $\mathbf{M}_0$, plotting β_r is challenging. Instead, we visualize the effects in a manner similar to the PC functions in Section 2.3.2; at each point we examine how strong the effect is in the orthogonal direction to the tangent plane (i.e. outward or inward relative to the face).

We provide three different plot types: directional plots, pointwise significance plots, and overall significance plots that control the Type 1 error rate simultaneously across the face. The directional plot shows how the beta function affects the face, the pointwise significance plot shows the facial areas where each point is tested positive (blue/red means positive at 99% level and lightblue/orange means positive at 95% level), and the overall significance plot shows the facial areas that have overall significance at 99% level. An advantage of applying functional data

analysis tools to faces are these overall significance plots, which rely heavily on the functional nature of the data.

We discuss only a few of the estimated effects here to highlight how to interpret our results. For example, the middle plot of Figure 2.11 presents the effect for sex, demonstrating the difference between the average male and female face, for a subject that is 30 years old, 170cm tall, and weighs 70kg. Blue denotes an inward effect, while red represents an outward effect. The female face is rounder than the male's as signified by the red parts around the cheek in the directional plot. It also shows that the male has a more pronounced nose, and the female has a rounder eye area with red on the eyelids and blue on the eyebrow area. Those areas are shown as significant for both pointwise significance plot and overall significance plot.

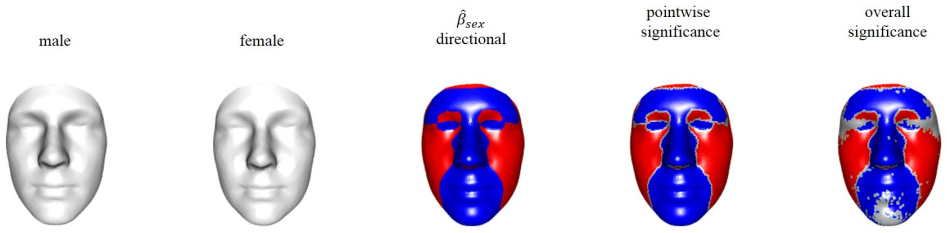

Figure 2.11: The left two plots are predicted faces of 30-year-old, 170cm-tall, 70kg-heavy Northern European male and female. The right three plots show the effect of beta of sex.

The effect of the proportion of East Asian descent is shown in Figure 2.12. As Northern European proportion is taken as the base, the beta function indicates the difference between Northern European and East Asian. We see that the average East Asian face is rounder, has a lower nose, and a less pronounced eyebrow. The overall significance plot (right most plot) shows that the nose, cheek, and forehead area are still statistically significant at a 99% significance level when correcting for

multiple testing across the entire face using our confidence bubbles.

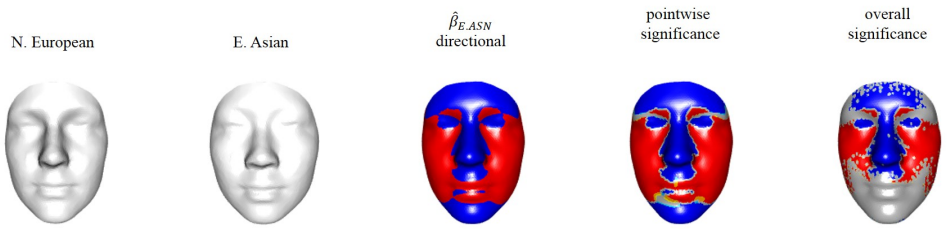

Figure 2.12: The left two plots are predicted faces of 25-year-old, 165cm-tall, 65kg-heavy Northern European and East Asian female. The right three plots show the effect of the corresponding beta.

The effect of the proportion of Western African is shown in Figure 2.13. The most features seem to be the nose and mouth, and those are picked up in the overall significance plot. The nose of Western African is more flattened but wider than that of Northern European, shown as the inward effect (blue) in the middle of nose, and the outward effect (red) on the sides of nose. The lips are more outward, and the lower cheek area difference is also picked up in the overall significance plot.

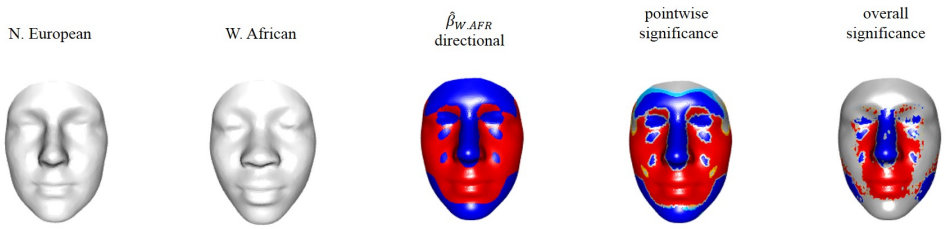

Figure 2.13: The left two plots are predicted faces of 25-year-old, 165cm-tall, 65kg-heavy Northern European and Western African female. The right three plots show the effect of beta of the corresponding beta.

2.4 Technical Proofs

2.4.1 2-Step Functional Principal Component Analysis

Step 1. Without loss of generality, assume that the $\mathbf{Y}_n(m)$ have been centered and thus have mean zero. Each of the functions is expressed as

$$\mathbf{Y}_n(m) = [Y_{n1}(m), \cdots, Y_{nD}(m)]^\top \approx \left[\sum_{j=1}^{J} \hat{b}_{nj1} e_j(m), \cdots, \sum_{j=1}^{J} \hat{b}_{njD} e_j(m)\right]^\top,$$

for $n = 1, \ldots, N$. We stack all of the coordinate-wise functions into a single vector of functions with dimension ND. We denote the resulting functions as $Y_l(m)$ where $Y_l(m) = Y_{nq}(m)$ for $l = N(q-1) + n$ and $n = 1, \cdots, N$, $q = 1, \cdots, D$.

We aim to find the pairs of eigenvalues η_h and principal component functions $\psi_h : \mathbf{M}_0 \to \mathbb{R}^D$, for $h = 1, \ldots, H$, which satisfy

$$\eta_h \psi_h(m) = \int \Phi(m, m') \psi_h(m') dm'$$

where

$$\begin{aligned}\mathbf{\Phi}(m, m') = E[Y_l(m) Y_l(m')^\top] &= \sum_{j_1}^{J} \sum_{j_2}^{J} E[b_{lj_1} b_{lj_2}] e_{j_1}(m) e_{j_2}(m')^\top \\ &= \sum_{j_1}^{J} \sum_{j_2}^{J} \Pi_{j_1, j_2} e_{j_1}(m) e_{j_2}(m')^\top,\end{aligned}$$

and $\|\psi_h\| = 1$.

We expand ψ_h using e_j:

$$\psi_h(m) = \sum_{j=1}^{J} w_{hj} e_j(m).$$

We then need to solve the following system of linear equations

$$\begin{aligned}\eta_h \sum_{j=1}^{J} w_{hj} e_j(m) &= \int_{\mathbf{M}_0} \left(\sum_{j_1}^{J} \sum_{j_2}^{J} \Pi_{j_1,j_2} e_{j_1}(m) e_{j_2}(m') \right) \left(\sum_{j_3}^{J} w_{hj_3} e_{j_3}(m') \right) dm' \\ &= \sum_{j_1}^{J} \sum_{j_2}^{J} \sum_{j_3}^{J} \Pi_{j_1,j_2} \left(\int_{\mathbf{M}_0} e_{j_2}(m') e_{j_3}(m') dm' \right) e_{j_1}(m) \\ &= \sum_{j_1}^{J} \sum_{j_2}^{J} \sum_{j_3}^{J} \Pi_{j_1,j_2} \mathbf{Z}_{j_2,j_3} e_{j_1}(m).\end{aligned}$$

And $\|\psi_h\| = 1$ means

$$\int_{\mathbf{M}_0} \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} w_{hj_1} e_{j_1}(m) w_{hj_2} e_{j_2}(m) dm = \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} w_{hj_1} w_{hj_2} \mathbf{Z}_{j_1,j_2} = 1.$$

Factor the matriz $\mathbf{Z} = \mathbf{G}^\top \mathbf{G}$ so that

$$\sum_{j_1=1}^{J} \sum_{j_2=1}^{J} w_{hj_1} w_{hj_2} \mathbf{Z}_{j_1,j_2} = \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} \sum_{j_3=1}^{J} w_{hj_1} w_{hj_2} \mathbf{G}_{j_1 j_3} \mathbf{G}_{j_3 j_2} = \sum_{j=1}^{J} a_{hj}^2,$$

where we define $a_{hj} = \sum_{j_1=1}^{J} w_{hj_1} \mathbf{G}_{jj_1}$. We then have

$$\begin{aligned}\sum_{j_1}^{J} \sum_{j_2}^{J} \sum_{j_3}^{J} \Pi_{j_1,j_2} \mathbf{Z}_{j_2,j_3} e_{j_1}(m) &= \sum_{j_1}^{J} \sum_{j_2}^{J} \sum_{j_3}^{J} \sum_{j_4}^{J} \Pi_{j_1 j_2} \mathbf{G}_{j_2 j_4} \mathbf{G}_{j_4 j_3} w_{hj_3} e_{j_1}(m) \\ &= \sum_{j_1}^{J} \sum_{j_2}^{J} \sum_{j_4}^{J} \Pi_{j_1 j_2} \mathbf{G}_{j_2 j_4} a_{hj_4} e_{j_1}(m) \\ &= \sum_{j_1}^{J} \sum_{j_4}^{J} \left(\sum_{j_2}^{J} \Pi_{j_1 j_2} \mathbf{G}_{j_2 j_4} \right) a_{hj_4} e_{j_1}(m).\end{aligned}$$

So we obtain the relation

$$\eta_h w_{hj_1} = \sum_{j_2}^{J} \left(\sum_{j_3}^{J} \Pi_{j_1 j_3} \mathbf{G}_{j_3 j_2} \right) a_{hj_2}$$

and

$$\eta_h a_{hj} = \eta_h \sum_{j_1}^{J} w_{hj_1} \mathbf{G}_{j,j_1} = \sum_{j_1}^{J} \sum_{j_2}^{J} \mathbf{G}_{j,j_1} \left(\sum_{j_3}^{J} \Pi_{j_1 j_3} \mathbf{G}_{j_3 j_2} \right) a_{hj_2} = \sum_{j_2}^{J} \tilde{\Pi}_{j,j_2} a_{hj_2},$$

where $\tilde{\Pi}_{j,j_2} = \sum_{j_1}^{J} \sum_{j_3}^{J} \mathbf{G}_{j,j_1} \Pi_{j_1 j_3} \mathbf{G}_{j_3 j_2}$. So we the vector $\mathbf{a}_j = \{a_{hj}\}$ is the j^{th} eigenvector of the covariance matrix $\tilde{\Pi}$. Since we know

$$a_{hj} = \sum_{j_1=1}^{J} w_{hj_1} \mathbf{G}_{jj_1},$$

reversing it would give w_{hj_1} and then we can get

$$\psi_h(m) = \sum_{j=1}^{J} w_{hj} e_j(m).$$

Step 2. We now expand $Y_{nq}(m)$ using the $\{\psi_h(m)\}$:

$$Y_{nq}(m) = \sum_{h=1}^{H} c_{nhq} \psi_h(m).$$

The coefficients $\mathbf{c} = \{c_{nhq}\}$ form an $N \times H \times D$ array. The covariance operator of $\mathbf{Y}_n(m)$ is given by

$$\begin{aligned} \Gamma_{q,q'}(m, m') = E[Y_{nq}(m) Y_{nq'}(m')] &= \sum_{h_1=1}^{H} \sum_{h_2=1}^{H} E[c_{nh_1q} c_{nh_2q'}^{\top}] \psi_{h_1}(m) \psi_{h_2}(m') \\ &= \sum_{h_1=1}^{H} \sum_{h_2=1}^{H} \Sigma_{h_1 q h_2 q'} \psi_{h_1}(m) \psi_{h_2}(m'). \end{aligned}$$

Now we find the pairs of eigenvalues λ_k and eigenfunctions $\mathbf{V}_k(m)$ that satisfy

$$\lambda_k \mathbf{V}_k(m) = \int \mathbf{\Gamma}(m, m') \mathbf{V}_k(m') dm'$$

where $\|\mathbf{V}_k\| = 1$. We expand $\mathbf{V}_k$ using the ψ_h as well:

$$V_{kq}(m) = \sum_{h=1}^{H} v_{khq}\psi_h(m),$$

where $\mathbf{v} = \{v_{khq}\}$ is a $K \times H \times D$ array of coefficients. So we want to solve

$$\begin{aligned}
\lambda_k \sum_{h=1}^{H} v_{khq}\psi_h(m) &= \sum_{q'=1}^{D} \int_{\mathbf{M}_0} \sum_{h_1=1}^{H}\sum_{h_2=1}^{H} \Sigma_{h_1qh_2q'}\psi_{h_1}(m)\psi_{h_2}(m') \sum_{h_3=1}^{H} v_{kh_3q'}\psi_{h_3}(m')dm' \\
&= \sum_{q'=1}^{D}\sum_{h_1=1}^{H}\sum_{h_2=1}^{H}\sum_{h_3=1}^{H} \Sigma_{h_1qh_2q'}\left(\int_{\mathbf{M}_0} \psi_{h_2}(m')\psi_{h_3}(m')dm'\right) v_{kh_3q'}\psi_{h_1}(m) \\
&= \sum_{q'=1}^{D}\sum_{h_1=1}^{H}\sum_{h_2=1}^{H}\sum_{h_3=1}^{H} \Sigma_{h_1qh_2q'}\mathbf{W}_{h_2h_3}v_{kh_3q'}\psi_{h_1}(m) \\
&= \sum_{q'=1}^{D}\sum_{h_1=1}^{H}\sum_{h_2=1}^{H} \Sigma_{h_1qh_2q'}v_{kh_2q'}\psi_{h_1}(m),
\end{aligned}$$

since $\mathbf{W}$ is the identity matrix as the ψ_h are orthogonal. Since $\|\mathbf{V}_k\| = 1$ this means that

$$\sum_{q=1}^{D}\sum_{h_1=1}^{H}\sum_{h_2=1}^{H} v_{kh_1q}v_{kh_2q} = 1.$$

Therefore

$$\lambda_k v_{khq} = \sum_{q'=1}^{D}\sum_{h_2=1}^{H} \Sigma_{hqh_2q'}v_{kh_2q'}.$$

So we have that $\mathbf{v}_k = \{v_{khq}\}$ is the k^{th} eigenmatrix of the $H \times D \times H \times D$ covariance tensor Σ.

2.4.2 Manifold-on-Scalar Regression with Regularization

Our objective is to find β's that minimizes

$$\sum_{n=1}^{N}\int_{\mathbf{M}_0}\left|\mathbf{Y}_n(m) - \sum_{r=1}^{R} x_{nr}\beta_r(m)\right|^2 dm + \sum_{r=1}^{R}\lambda_r\int_{\mathbf{M}_0}|L\beta_r(m)|^2 dm \qquad (2.4.1)$$

We can take $\lambda = \lambda_1 = \cdots = \lambda_R$, but we will keep them separate for now.

We need to choose roughness operator L carefully. Since we do not want the minimizer of (2.4.1) to change depending on the coordinate system, we need an operator that is invariant to rotation and translation. Ramsey (2002) chooses Laplacian operator as such operator. Laplacian operator $\triangle$ is such that $\triangle f = f_{xx} + f_{yy}$.

Wtih $\mathbf{f} : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}^3$, the Laplacian operator on $\mathbf{f}$ is as

$$\begin{aligned}\triangle \mathbf{f} = \triangle [f_1, f_2, f_3]^\top &= [\triangle f_1, \triangle f_2, \triangle f_3]^\top \\ &= \left[\frac{d^2 f_1}{dm_1^2} + \frac{d^2 f_1}{dm_2^2}, \frac{d^2 f_2}{dm_2^2} + \frac{d^2 f_2}{dm_1^2}, \frac{d^2 f_3}{dm_1^2} + \frac{d^2 f_3}{dm_2^2}\right]^\top\end{aligned}$$

where f_1, f_2, f_3 correspond to each coordinate of $\mathbf{f}$ and m_1 and m_2 correspond to each coordinate of $\mathbf{M}_0$.

Therefore (2.4.1) becomes

$$\sum_{n=1}^{N} \int_{\mathbf{M}_0} \left| \mathbf{Y}_n(m) - \sum_{r=1}^{R} x_{nr} \beta_r(m) \right|^2 dm + \sum_{r=1}^{R} \lambda_r \int_{\mathbf{M}_0} |\triangle \beta_r(m)|^2 dm \qquad (2.4.2)$$

With PC basis functions $\mathbf{V}_k : [\mathbf{M}_0 \subset \mathbb{R}^2] \to \mathbb{R}^3$ for $k = 1, \cdots, K$, $\mathbf{Y}_n$ and β_r can both be expanded with $\mathbf{V}_1, \cdots, \mathbf{V}_K$:

$$\begin{aligned}\sum_{n=1}^{N} \int_{\mathbf{M}_0} \left| \sum_{k=1}^{K} y_{nk} \mathbf{V}_k(m) - \sum_{r=1}^{R} x_{nr} \sum_{k=1}^{K} b_{rk} \mathbf{V}_k(m) \right|^2 dm \\ + \sum_{r=1}^{R} \lambda_r \int_{\mathbf{M}_0} \left| \sum_{k=1}^{K} b_{rk} (\triangle \mathbf{V}_k(m)) \right|^2 dm \qquad (2.4.3)\end{aligned}$$

The first term

$$
\begin{aligned}
&\sum_{n=1}^{N}\int_{\mathbf{M}_0}\left|\sum_{k=1}^{K}y_{nk}\mathbf{V}_k-\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right|^2 dm\\
&=\sum_{n=1}^{N}\int_{\mathbf{M}_0}\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k-\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)^{\top}\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k-\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)dm\\
&=\sum_{n=1}^{N}\int_{\mathbf{M}_0}\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k\right)^{\top}\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k\right)-\left(\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)^{\top}\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k\right)\\
&\quad-\left(\sum_{k=1}^{K}y_{nk}\mathbf{V}_k\right)^{\top}\left(\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)+\left(\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)^{\top}\left(\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}\mathbf{V}_k\right)dm\\
&=\sum_{n=1}^{N}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}y_{nk_1}y_{nk_2}\int_{\mathbf{M}_0}\mathbf{V}_{k_1}^{\top}\mathbf{V}_{k_2}dm-2\sum_{n=1}^{N}\sum_{r=1}^{R}x_{nr}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}b_{rk_1}y_{nk_2}\int_{\mathbf{M}_0}\mathbf{V}_{k_1}^{\top}\mathbf{V}_{k_2}dm\\
&\quad+\sum_{n=1}^{N}\sum_{r_1=1}^{R}\sum_{r_2=1}^{R}x_{nr_1}x_{nr_2}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}b_{r_1k_1}b_{r_2k_2}\int_{\mathbf{M}_0}\mathbf{V}_{k_1}^{\top}\mathbf{V}_{k_2}dm\\
&=\sum_{n=1}^{N}\sum_{k=1}^{K}y_{nk}^2-2\sum_{n=1}^{N}\sum_{r=1}^{R}x_{nr}\sum_{k=1}^{K}b_{rk}y_{nk}+\sum_{n=1}^{N}\sum_{r_1=1}^{R}\sum_{r_2=1}^{R}x_{nr_1}x_{nr_2}\sum_{k=1}^{K}b_{r_1k}b_{r_2k}\\
&=\sum_{n=1}^{N}\sum_{k=1}^{K}\left(y_{nk}^2-2\sum_{r=1}^{R}x_{nr}b_{rk}y_{nk}+\sum_{r_1=1}^{R}\sum_{r_2=1}^{R}x_{nr_1}x_{nr_2}b_{r_1k}b_{r_2k}\right)^2\\
&=\sum_{n=1}^{N}\sum_{k=1}^{K}\left(y_{nk}-\sum_{r=1}^{R}x_{nr}b_{rk}\right)^2
\end{aligned}
$$

The second term

$$
\begin{aligned}
&\sum_{r=1}^{R}\lambda_r\int_{\mathbf{M}_0}\left|\sum_{k=1}^{K}b_{rk}(\triangle\mathbf{V}_k)\right|^2 dm\\
&=\sum_{r=1}^{R}\lambda_r\int_{\mathbf{M}_0}\left(\sum_{k=1}^{K}b_{rk}(\triangle\mathbf{V}_k)\right)^{\top}\left(\sum_{k=1}^{K}b_{rk}(\triangle\mathbf{V}_k)\right)dm\\
&=\sum_{r=1}^{R}\lambda_r\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}b_{rk_1}b_{rk_2}\int_{\mathbf{M}_0}(\triangle\mathbf{V}_{k_1})^{\top}(\triangle\mathbf{V}_{k_2})dm\\
&=\sum_{r=1}^{R}\lambda_r\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}b_{rk_1}b_{rk_2}U_{k_1,k_2}
\end{aligned}
$$

where

$$
U_{k_1,k_2}=\int_{\mathbf{M}_0}(\triangle\mathbf{V}_{k_1})^{\top}(\triangle\mathbf{V}_{k_2})dm
$$

$$= \int_{\mathbf{M}_0} (\triangle V_{k_1,1})^2 + (\triangle V_{k_1,2})^2 + (\triangle V_{k_1,3})^2 dm.$$

that is summing the three coordinates.

And from FPCA, we know

$$V_{kq}(m) = \sum_{h=1}^{H} v_{khq}\phi_h(m)$$

and

$$\phi_h(m) = \sum_{j=1}^{J} w_{hj}e_j(m)$$

where e_j's are the felsplines (Ramsey, 2002).

Then

$$\int_{\mathbf{M}_0} (\triangle V_{k,q}(m))^2 dm = \int_{\mathbf{M}_0} \left(\sum_{h=1}^{H} (\triangle \phi_h(m) \right)^2 dm.$$

And in order to get $\int_{\mathbf{M}_0} (\triangle \phi_h(m))^2 dm$, we need to consider the FEM theories.

Let $f_h(m) = - \triangle \phi_h(m)$.

$$\begin{aligned}
\langle f_h, e_j \rangle &= \int_{\mathbf{M}_0} (- \triangle \phi_h) e_j dm \\
&= \int_{\mathbf{M}_0} (\bigtriangledown \phi_h)(\bigtriangledown e_j) \quad (\because \text{ Green's theorem}) \\
&= \int_{\mathbf{M}_0} (\bigtriangledown \sum_{j_1=1}^{J} w_{hj_1} e_{j_1})(\bigtriangledown e_j) \\
&= \sum_{j_1=1}^{J} w_{hj_1} \int_{\mathbf{M}_0} (\bigtriangledown e_{j_1})(\bigtriangledown e_j) dm
\end{aligned}$$

And we have code for getting $\int_{\mathbf{M}_0} (\bigtriangledown e_{j_1})(\bigtriangledown e_j) dm$. The matrix with these components is called stiffness matrix.

Then using

$$f_h(m) = \sum_{j=1}^{J} \langle f_h, e_j \rangle e_j(m) = \sum_{j=1}^{J} f_{hj} e_j(m),$$

we can get

$$\begin{aligned}
\int_{\mathbf{M}_0} (\triangle \phi_h)^2 dm &= \int_{\mathbf{M}_0} f_h(m)^2 dm \\
&= \int_{\mathbf{M}_0} (\sum_{j_1=1}^{J} f_{hj_1} e_{j_1}(m))(\sum_{j_2=1}^{J} f_{hj_2} e_{j_2}(m)) dm \\
&= \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} f_{hj_1} f_{hj_2} \int_{\mathbf{M}_0} e_{j_1}(m) e_{j_2}(m) dm.
\end{aligned}$$

The matrix with components of the inner product between e_{j_1} and e_{j_2} ($\int_{\mathbf{M}_0} e_{j_1}(m) e_{j_2}(m) dm$) is called mass matrix, and we have code for that too.

Let's go back to getting the least square estimate of $\{b_{rk}\}$.

Let

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1K} \\ y_{21} & y_{22} & \cdots & y_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NK} \end{bmatrix}, \quad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1R} \\ x_{21} & x_{22} & \cdots & x_{2R} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NR} \end{bmatrix},$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ b_{21} & b_{22} & \cdots & b_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ b_{R1} & b_{R2} & \cdots & b_{RK} \end{bmatrix}, \quad \Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_R).$$

Then our objective becomes to find B that minimizes

$$trace\{(Y - XB)^\top (Y - XB)\} + trace\{\Lambda B U B^\top\} \tag{2.4.4}$$

Now let's find the least square estimate of B.

Differentiate (2.4.4) and set it to 0:

$$-2X^\top Y + 2X^\top XB + 2\Lambda BU = 0.$$

We can cross 2 out:

$$-X^\top Y + X^\top XB + \Lambda BU = 0.$$

Take transpose of everything:

$$-Y^\top X + B^\top (X^\top X) + U^\top B^\top \Lambda = 0.$$

Vectorize the whole thing:

$$-\text{vec}(Y^\top X) + ((X^\top X) \otimes I_K)\text{vec}(B^\top) + (\Lambda \otimes U^\top)\text{vec}(B^\top) = 0.$$

Then the least square estimate of B is:

$$\text{vec}(\hat{B}^\top) = \left((X^\top X) \otimes I_K + \Lambda \otimes U^\top\right)^{-1} \text{vec}(Y^\top X). \tag{2.4.5}$$

Find covariance of $\text{vec}(\hat{B}^\top)$. Let

$$A = \left((X^\top X) \otimes I_K + \Lambda \otimes U^\top\right)^{-1}.$$

$$\begin{aligned}
\text{cov}\left(\text{vec}(\hat{B}^\top)\right) &= A\text{cov}\left(\text{vec}(Y^\top X)\right) A^\top \\
&= A\text{cov}\left((X^\top \otimes I_K)\text{vec}(Y^\top)\right) A^\top \\
&= A(X^\top \otimes I_K)(I_N \otimes \Sigma)(X \otimes I_K)A^\top
\end{aligned}$$

$$= A(X^\top \otimes \Sigma)(X \otimes I_K)A^\top$$
$$= A\left((X^\top X) \otimes \Sigma\right)A^\top$$

2.4.3 Proof of Theorem 2.1.1

Using the Karhunen-Loève (KL) expansion, we can write

$$\sqrt{N}(\hat{\beta}_r(m) - \beta_r(m)) = \sum_{j=1}^{\infty} \sqrt{\lambda_j} Z_j \mathbf{U}_j(m),$$

where the equality hold for almost all $m \in \mathbf{M}_0$ since $L^2(\mathbf{M}_0)$ consists of equivalence classes. By the Cauchy-Schwarz inequality,

$$\begin{aligned} N|\hat{\beta}_r(m) - \beta_r(m)|^2 &\leq \sum_{j=1}^{\infty} |\lambda_j^{\frac{1}{4}} Z_j|^2 \sum_{j=1}^{\infty} |\lambda_j^{\frac{1}{4}} \mathbf{U}_j(m)|^2 \\ &= \sum_{j=1}^{\infty} \sqrt{\lambda_j} Z_j^2 \sum_{j=1}^{\infty} \sqrt{\lambda_j} |\mathbf{U}_j(m)|^2, \end{aligned}$$

as desired.

Chapter 3

Manifold-on-Scalar Regression Algorithms

3.1 Kernel Expansion

In this Chapter, we present the three algorithms that are for estimating the parameter functions in manifold-on-scalar regression. Instead of using felspline expansion in Chapter 2, we now use kernel expansion to embed the sample of manifolds in a Reproducing Kernel Hilbert Space (RKHS). We used an exponential kernel that is of the form

$$K(u, u') = exp(-\sigma \|u - u'\|).$$

Exponential kernel is useful because it produces RKHS that is equivalent to a Sobolev space. Exponential kernel is also a special case of Matérn kernel with its smoothness parameter being 1/2.

Penalized least squares is used to find the coefficients for the data.

$$\sum_{n=1}^{N}\sum_{p=1}^{P}(Y_{np}-\sum_{j=1}^{J}c_{nj}K(u_p,u_j))^2+\lambda\left\|c_{nj}K(u_p,u_j)\right\|_{\mathbb{K}}^2$$

We tried a few different pairs of σ for the exponential kernel and λ and found $\sigma = 3.5$ and $\lambda = 10^{-4}$ give the lowest generalized cross-validation value. The resulting manifolds would be

$$\begin{aligned}\mathbf{Y}_n(m) &= [Y_{n1}(m),\cdots,Y_{nD}(m)]^\top \\ &\approx \left[\sum_{j=1}^{J}\hat{a}_{nj1}K(m_j,m),\cdots,\sum_{j=1}^{J}\hat{a}_{njD}K(m_j,m)\right]^\top.\end{aligned}$$

The fitted faces are shown in Figure 3.1. The fit to the original data is clearly much better with the kernel expansion than with felspline expansion. With felspline expansion, the area around eyes and noses are a bit smoothed out, but with kernel expansion, we are able to retain the features and curvatures of the original data.

3.2 Algorithms

We now present three different algorithms for parameter estimation in manifold-on-scalar regression. The first one is a principal component regression method, the second one is also a principal component regression but with smoothness imposed on the covariance operator, and the third is a penalized regression method.

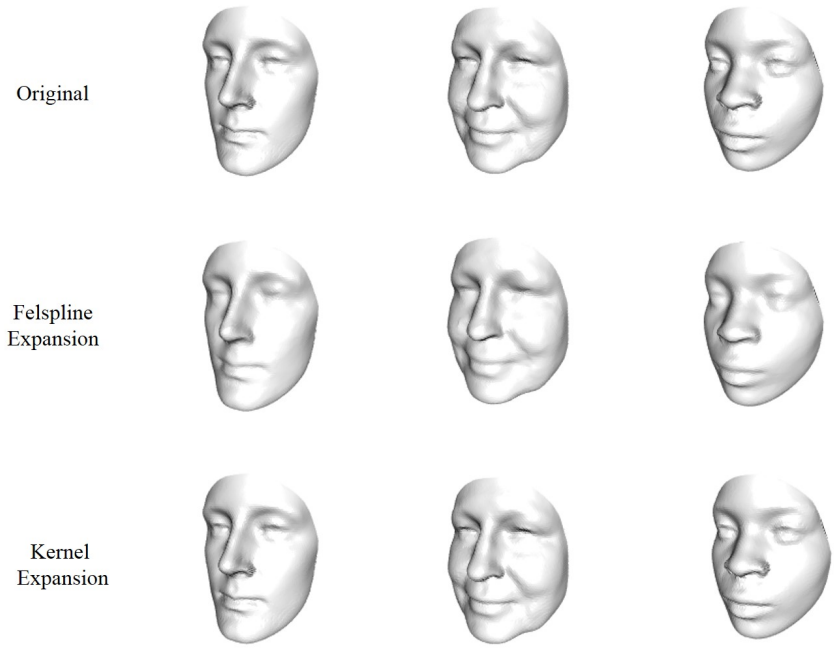

Figure 3.1: Example of faces with kernel eigenfunction expansion with comparison to faces expanded using felsplines.

3.2.1 Principal Component Regression

We conduct a principal component analysis on the kernel-expanded functional data and run a regression model on the principal components. Therefore, it is like a principal component regression.

First we would centralize $Y_{nq}(m)$ for $q = 1, \cdots, D$. The sample mean of $Y_{nq}(m)$ is

$$\hat{\mu}_q(m) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J}\hat{a}_{njq}K(m_j, m) = \sum_{j=1}^{J}\left(\frac{1}{N}\sum_{n=1}^{N}\hat{a}_{njq}\right)K(m_j, m),$$

Let $\tilde{Y}_{nq}(m)$ is the centralized $Y_{nq}(m)$. It is

$$\tilde{Y}_{nq}(m) = Y_{nq}(m) - \hat{\mu}_q(m) = \sum_{j=1}^{J}\left[\hat{a}_{njq} - \left(\frac{1}{N}\sum_{n=1}^{N}\hat{a}_{njq}\right)\right]K(m_j, m)$$

$$= \sum_{j=1}^{J} \tilde{a}_{njq} K(m_j, m).$$

The coefficients $\mathbf{a} = \{\tilde{a}_{njq}\}$ form an $N \times J \times D$ array. The covariance operator of $\mathbf{Y}_n(m)$ is given by

$$\begin{aligned}\mathbf{\Gamma}_{q,q'}(m, m') = E[\tilde{Y}_{nq}(m)\tilde{Y}_{nq'}(m')] &= \sum_{j_1=1}^{J}\sum_{j_2=1}^{J} E[\tilde{a}_{nj_1q}\tilde{a}_{nj_2q'}]K(m_{j_1}, m)K(m_{j_2}, m') \\ &= \sum_{j_1=1}^{J}\sum_{j_2=1}^{J} \Sigma_{j_1qj_2q'}K(m_{j_1}, m)K(m_{j_2}, m').\end{aligned}$$

Now we find the pairs of eigenvalues λ_k and eigenfunctions $\mathbf{V}_k(m)$ that satisfy

$$\lambda_k \mathbf{V}_k(m) = \int_{M_0} \mathbf{\Gamma}(m, m')\mathbf{V}_k(m')dm'$$

where $\|\mathbf{V}_k\| = 1$. We expand $\mathbf{V}_k$ using the $K(m_j, m)$ as well:

$$V_{kq}(m) = \sum_{j=1}^{J} v_{kjq} K(m_j, m),$$

where $\mathbf{v} = \{v_{kjq}\}$ is a $K \times J \times D$ array of coefficients. So we want to solve

$$\begin{aligned}&\lambda_k \sum_{j=1}^{J} v_{kjq} K(m_j, m) \\ &\quad= \sum_{q'=1}^{D} \int_{M_0} \sum_{j_1=1}^{J}\sum_{j_2=1}^{J} \Sigma_{j_1qj_2q'}K(m_{j_1}, m)K(m_{j_2}, m'), \sum_{j_3=1}^{J} v_{kj_3q'}K(m_{j_3}, m')dm' \\ &\quad= \sum_{q'=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J} \Sigma_{j_1qj_2q'} \int_{M_0} K(m_{j_2}, m'), K(m_{j_3}, m')dm' v_{kj_3q'}K(m_{j_1}, m) \\ &\quad= \sum_{q'=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J} \Sigma_{j_1qj_2q'}\mathbf{W}_{j_2,j_3} v_{kj_3q'}K(m_{j_1}, m),\end{aligned}$$

where $\mathbf{W}$ is $J \times J$ matrix of evaluated kernel inner product:

$$\mathbf{W}_{j_2,j_3} = \int_{\mathbf{M}_0} K(m_{j_2}, m)K(m_{j_3}, m)dm.$$

Since $\|\mathbf{V}_k\|_{L2} = 1$, this means that

$$\begin{aligned}\int_{M_0} \mathbf{V}_k^\top(m)\mathbf{V}_k(m)dm &= \sum_{q=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J} v_{kj_1q}v_{kj_2q}\int_{\mathbf{M}_0} K(m_{j_1}, m)K(m_{j_2}, m)dm \\ &= \sum_{q=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J} v_{kj_1q}v_{kj_2q}\mathbf{W}_{j_1,j_2} = 1.\end{aligned}$$

Factor the matrix $\mathbf{W} = \mathbf{G}^\top\mathbf{G}$ so that

$$\begin{aligned}\|\mathbf{V}_k\|_{L2} &= \sum_{j_1=1}^{J}\sum_{j_2=1}^{J} v_{kj_1q}v_{kj_2q}\mathbf{W}_{j_1,j_2} \\ &= \sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J} v_{kj_1q}v_{kj_2q}[\mathbf{G}^\top]_{j_1,j_3}\mathbf{G}_{j_3,j_2} \\ &= \sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J} v_{kj_1q}v_{kj_2q}\mathbf{G}_{j_3,j_1}\mathbf{G}_{j_3,j_2} \\ &= \sum_{j_3=1}^{J}\left(\sum_{j_1=1}^{J} v_{kj_1q}\mathbf{G}_{j_3,j_1}\right)\left(\sum_{j_2=1}^{J} v_{kj_2q}\mathbf{G}_{j_3,j_2}\right) \\ &= \sum_{j=1}^{J} b_{kjq}^2,\end{aligned}$$

where we define $b_{kjq} = \sum_{j_1=1}^{J} v_{kj_1q}\mathbf{G}_{jj_1}$. We then have

$$\begin{aligned}&\sum_{q'=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J}\Sigma_{j_1qj_2q'}\mathbf{W}_{j_2,j_3}v_{kj_3q'}K(m_{j_1}, m) \\ &\qquad = \sum_{q'=1}^{D}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J}\sum_{j_4=1}^{J}\Sigma_{j_1qj_2q'}\mathbf{G}_{j_4,j_2}\mathbf{G}_{j_4,j_3}v_{kj_3q'}K(m_{j_1}, m)\end{aligned}$$

$$= \sum_{q'=1}^{D} \sum_{j_1=1}^{J} \sum_{j_4=1}^{J} \left(\sum_{j_2=1}^{J} \Sigma_{j_1 q j_2 q'} \mathbf{G}_{j_4,j_2} \right) b_{k j_4 q'} K(m_{j_1}, m)$$

So we obtain the relation

$$\lambda_k v_{k j_1 q} = \sum_{q'=1}^{D} \sum_{j_4=1}^{J} \left(\sum_{j_2=1}^{J} \Sigma_{j_1 q j_2 q'} \mathbf{G}_{j_4,j_2} \right) b_{k j_4 q'}$$

and

$$\begin{aligned}
\lambda_k b_{kjq} &= \lambda_k \sum_{j_1=1}^{J} v_{k j_1 q} \mathbf{G}_{j j_1} \\
&= \sum_{j_1=1}^{J} \left[\sum_{q'=1}^{D} \sum_{j_4=1}^{J} \left(\sum_{j_2=1}^{J} \Sigma_{j_1 q j_2 q'} \mathbf{G}_{j_4,j_2} \right) b_{k j_4 q'} \right] \mathbf{G}_{j j_1} \\
&= \sum_{q'=1}^{D} \sum_{j_4=1}^{J} \left[\sum_{j_1=1}^{J} \sum_{j_2=1}^{J} \mathbf{G}_{j j_1} \left(\Sigma_{j_1 q j_2 q'} \right) \mathbf{G}_{j_4,j_2} \right] b_{k j_4 q'} \\
&= \sum_{q'=1}^{D} \sum_{j_4=1}^{J} \tilde{\Sigma}_{j q j_4 q'} b_{k j_4 q'}
\end{aligned}$$

where $\tilde{\Sigma}_{j q j_4 q'} = \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} \mathbf{G}_{j j_1} \left(\Sigma_{j_1 q j_2 q'} \right) \mathbf{G}_{j_4,j_2}$.
So we have that $\mathbf{b}_k = \{b_{kjq}\}$ is the k^{th} eigenmatrix of $J \times D \times J \times D$ covariance tensor $\tilde{\Sigma}$.

As $b_{kjq} = \sum_{j_1=1}^{J} v_{k j_1 q} \mathbf{G}_{j j_1}$, reversing it would give us $v_{k j_1 q}$ and the eigenfunctions $V_{kq}(m) = \sum_{j=1}^{J} v_{kjq} K(m_j, m)$. We can write

$$\mathbf{Y}_n(m) = \sum_{k=1}^{K} c_{nk} \mathbf{V}_k(m).$$

3.2.2 Principal Component Regression with RKHS penalty on Covariance Operator

We conduct a principal component regression in this algorithm also, but the difference between this algorithm and the algorithm in Section 3.2.1 is that for this we impose smoothness on the covariance operator. The covariance operator of $\mathbf{Y}_n(m)$ in Section 3.2.1 was

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \left\{ \sum_{n=1}^{N} \|\tilde{Y}_n \otimes \tilde{Y}_n - \mathbf{\Gamma}\|_{L2}^2 \right\}.$$

We look at this $\mathbf{\Gamma}$ coordinate-wise and put RKHS penalty on $\mathbf{\Gamma}_{q,q'}$.
The estimator of covariance operator $\hat{\mathbf{\Gamma}}_{q,q'}$ is now the one that minimizes

$$\frac{1}{N} \sum_{n=1}^{N} \|\tilde{Y}_{nq} \cdot \tilde{Y}_{nq'} - \mathbf{\Gamma}_{q,q'}\|_{L2}^2 + \lambda \|\mathbf{\Gamma}_{q,q'}\|_{H_K}^2. \tag{3.2.1}$$

We expand $\mathbf{\Gamma}_{q,q'} : \mathbf{M}_0 \times \mathbf{M}_0 \to \mathbb{R}$ with tensor of kernel functions.

$$\mathbf{\Gamma}_{q,q'}(m, m') = \sum_{k=1}^{K} \sum_{l=1}^{L} \gamma_{kqlq'} K(m_k, m) K(m_l, m').$$

Therefore, now we want to find $\{\gamma_{kl}\}$ that minimizes equation (3.2.1) which becomes

$$\frac{1}{N} \sum_{n=1}^{N} \left\| \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} \tilde{a}_{nj_1q} \tilde{a}_{nj_2q} K_{j_1}(m) K_{j_2}(m') - \sum_{k=1}^{K} \sum_{l=1}^{L} \gamma_{kqlq'} K_{m_k}(m) K_{m_l}(m') \right\|_{L2}^2 + \lambda \left\| \sum_{k=1}^{K} \sum_{l=1}^{L} \gamma_{kqlq'} K(m_k, m) K(m_l, m') \right\|_{H_K}^2.$$

This is

$$\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{j_3=1}^{J}\sum_{j_4=1}^{J}\Sigma_{j_1qj_2q'}\Sigma_{j_3qj_4q'}\int K_{j_1}(m)K_{j_3}(m)dm\int K_{j_2}(m')K_{j_4}(m')dm'$$
$$-2\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\Sigma_{j_1qj_2q'}\gamma_{kqlq'}\int K_{j_1}(m)K_{k}(m)dm\int K_{j_2}(m')K_{l}(m')dm'$$
$$+\sum_{k_1=1}^{K}\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\sum_{l_2=1}^{L}\gamma_{k_1q,l_1q'}\gamma_{k_2q,l_2q'}\int K_{k_1}(m)K_{k_2}(m)dm\int K_{l_1}(m')K_{l_2}(m')dm'$$
$$+\lambda\sum_{k_1=1}^{K}\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\sum_{l_2=1}^{L}\gamma_{k_1q,l_1q'}\gamma_{k_2q,l_2q'}K(m_{k_1},m_{k_2})K(m_{l_1},m_{l_2}).$$

Let W_{j_1,j_2} be a matrix of evaluated $\int K_{j_1}(m)K_{j_2}(m)dm$ for the sets (m_{j_1},m_{j_2}). Then the above, excluding the ones that does not depend on $\gamma_{kqlq'}$, becomes

$$-2\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\Sigma_{j_1qj_2q'}\gamma_{kqlq'}W_{j_1,k}W_{j_2,l}$$
$$+\sum_{k_1=1}^{K}\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\sum_{l_2=1}^{L}\gamma_{k_1q,l_1q'}\gamma_{k_2q,l_2q'}W_{k_1,k_2}W_{l_1,l_2}$$
$$+\lambda\sum_{k_1=1}^{K}\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\sum_{l_2=1}^{L}\gamma_{k_1q,l_1q'}\gamma_{k_2q,l_2q'}K(m_{k_1},m_{k_2})K(m_{l_1},m_{l_2}).$$

Since we look at this for each (q,q'), let Σ_{j_1,j_2} be the matrix corresponding to $\Sigma_{j_1,q,j_2,q'}$ for a fixed (q,q') and let γ_{kl} represent $\gamma_{kqlq'}$.

Let

$$A_{kl}=\sum_{j_1}\sum_{j_2}W_{j_1k}\Sigma_{j_1,j_2}W_{j_2,l}.$$

Then the first line becomes

$$-2\sum_{j_1=1}^{J}\sum_{j_2=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\Sigma_{j_1j_2}\gamma_{kl}W_{j_1,k}W_{j_2,l}=-2\sum_{k=1}^{K}\sum_{l=1}^{L}\gamma_{kl}A_{kl}=-2\,\mathrm{trace}\{\gamma^{\top}A\}.$$

The second line is

$$\sum_{k_2=1}^{K}\sum_{l_1=1}^{L}\left[\sum_{k_1=1}^{K}\gamma_{k_1,l_1}W_{k_1,k_2}\right]\left[\sum_{l_2=1}^{L}\gamma_{k_2,l_2}W_{l_1,l_2}\right]=\operatorname{trace}\left\{\left[\gamma^\top W^k\right]_{l_1,k_2}\left[\gamma(W^l)^\top\right]_{k_2,l_1}\right\}.$$

And the third line is

$$\lambda\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\left[\sum_{k_1=1}^{K}\gamma_{k_1,l_1}K(m_{k_1},m_{k_2})\right]\left[\sum_{l_2=1}^{L}\gamma_{k_2,l_2}K(m_{l_1},m_{l_2})\right]$$
$$=\lambda\operatorname{trace}\left\{\left[\gamma^\top K^k\right]_{l_1,k_2}\left[\gamma(K^l)^\top\right]_{k_2,l_1}\right\}.$$

Therefore, we want to find γ that minimizes

$$-2\operatorname{trace}\{\gamma^\top A\}+\operatorname{trace}\left\{\left[\gamma^\top W^k\right]\left[\gamma(W^l)^\top\right]\right\}+\lambda\operatorname{trace}\left\{\left[\gamma^\top K^k\right]\left[\gamma(K^l)^\top\right]\right\}.$$

Differentiate by γ would yield:

$$-2A+W^k\gamma(W^l)^\top+(W^k)^\top\gamma W^l+\lambda K^k\gamma(K^l)^\top+\lambda(K^k)^\top\gamma K^l=0.$$

Vectorize the whole thing:

$$(W^l\otimes W^k)vec(\gamma)+(W^l\otimes W^k)^\top vec(\gamma)+\lambda(K^l\otimes K^k)vec(\gamma)+\lambda(K^l\otimes K^k)^\top vec(\gamma)$$
$$=2vec(A).$$

Therefore,

$$vec(\hat{\gamma})=2\left\{(W^l\otimes W^k)+(W^l\otimes W^k)^\top+\lambda(K^l\otimes K^k)+\lambda(K^l\otimes K^k)^\top\right\}^{-1}vec(A).$$

If we assume that we have taken $W^k=W^l=W$ and $K^k=K^l=K$, meaning that

we have taken same points $(k_1, k_2) = (l_1, l_2)$, and W and K are symmetric, then

$$vec(\hat{\gamma}) = \{(W \otimes W) + \lambda(K \otimes K)\}^{-1} vec(A). \tag{3.2.2}$$

Then we rearrange $vec(\hat{\gamma})$ to get $\hat{\gamma}$ and conduct singular value decomposition on this $\hat{\gamma}$ to get the eigenfunctions. We can then use the coefficients for those eigenfunctions in regression.

3.2.3 Penalized Regression with RKHS Penalty

This time we do not conduct a principal component analysis but we impose an RKHS penalty term on the β functions. Our objective is to find β's that minimizes

$$\ell_\lambda(\boldsymbol{\beta}) = \sum_{n=1}^{N} \left\| \mathbf{Y}_n(\cdot) - \sum_{r=1}^{R} x_{nr} \boldsymbol{\beta}_r(\cdot) \right\|_{L2}^2 + \sum_{r=1}^{R} \lambda_r \|\boldsymbol{\beta}_r(\cdot)\|_{H_K}^2 \tag{3.2.3}$$

$\mathbf{Y}_n$ is expanded as:

$$\mathbf{Y}_n(m) = [Y_{n1}(m), \cdots, Y_{nD}(m)]^\top \approx \left[\sum_{j=1}^{J} \hat{a}_{nj1} K(m_j, m), \cdots, \sum_{j=1}^{J} \hat{a}_{njD} K(m_j, m) \right]^\top.$$

The estimator $\hat{\boldsymbol{\beta}}$ then minimizes $\ell_\lambda(\boldsymbol{\beta})$. We take the expansion of $\boldsymbol{\beta}_r$ as:

$$\boldsymbol{\beta}_r(m) = [\beta_{r1}(m), \cdots, \beta_{rD}(m)]^\top = \left[\sum_{l=1}^{L} b_{rl1} K(m_l, m), \cdots, \sum_{l=1}^{L} b_{rlD} K(m_l, m) \right]^\top.$$

So the goal becomes to find $\{\hat{b}_{rlq}\}$ that minimizes $\ell_\lambda(\boldsymbol{\beta})$. It is possible to have $L = J$ and $\{m_l\} = \{m_j\}$. But let's keep them separate for now.

We again look at this coordinate-wise. Then for each $q = 1, \cdots, D$, the estimator

$\hat{\boldsymbol{\beta}}_{rq}$ is the one that minimizes

$$\ell_\lambda(\boldsymbol{\beta}) = \sum_{n=1}^{N} \left\| \mathbf{Y}_{nq}(\cdot) - \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot) \right\|_{L2}^2 + \sum_{r=1}^{R} \lambda_r \|\boldsymbol{\beta}_{rq}(\cdot)\|_{H_K}^2. \tag{3.2.4}$$

The first part of (3.2.4) is

$$\begin{aligned}
&\sum_{n=1}^{N} \left\| \mathbf{Y}_{nq}(\cdot) - \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot) \right\|_{L2}^2 \\
&\quad = \sum_{n=1}^{N} \left\langle \mathbf{Y}_{nq}(\cdot) - \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot), \mathbf{Y}_{nq}(\cdot) - \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot) \right\rangle_{L2} \\
&\quad = \sum_{n=1}^{N} \langle \mathbf{Y}_{nq}(\cdot), \mathbf{Y}_{nq}(\cdot) \rangle_{L2} - 2\sum_{n=1}^{N} \left\langle \mathbf{Y}_{nq}(\cdot), \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot) \right\rangle_{L2} \\
&\qquad\qquad + \sum_{n=1}^{N} \left\langle \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot), \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_{rq}(\cdot) \right\rangle_{L2}.
\end{aligned}$$

Excluding the ones that do not depend on $\boldsymbol{\beta}$,

$$\begin{aligned}
&\sum_{n=1}^{N} \left[-2 \left\langle \mathbf{Y}_n(\cdot), \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_r(\cdot) \right\rangle_{L2} + \left\langle \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_r(\cdot), \sum_{r=1}^{R} x_{nr}\boldsymbol{\beta}_r(\cdot) \right\rangle_{L2} \right] \\
&\quad = -2\sum_{n=1}^{N}\sum_{j=1}^{J}\sum_{l=1}^{L}\sum_{r=1}^{R} \hat{a}_{njq} b_{rlq} \int K(m_j, m)K(m_l, m)dm \\
&\qquad + \sum_{n=1}^{N}\sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L} x_{nr_1} b_{r_1 l_1 q} x_{nr_2} b_{r_2 l_2 q} \int K(m_{l_1}, m)K(m_{l_2}, m)dm.
\end{aligned}$$

The second part of (3.2.4) is

$$\sum_{r=1}^{R} \lambda_r \|\boldsymbol{\beta}_{rq}(\cdot)\|_{H_K}^2 = \sum_{r=1}^{R} \lambda_r \sum_{l_1=1}^{L}\sum_{l_2=1}^{L} b_{rl_1q} b_{rl_2q} K(m_{l_1}, m_{l_2}).$$

Let

$$A(q) = \begin{bmatrix} \hat{a}_{11q} & \hat{a}_{12q} & \cdots & \hat{a}_{1Jq} \\ \hat{a}_{21q} & \hat{a}_{22q} & \cdots & \hat{a}_{2Jq} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{a}_{N1q} & \hat{a}_{N2q} & \cdots & \hat{a}_{NJq} \end{bmatrix}_{N\times J}, \quad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1R} \\ x_{21} & x_{22} & \cdots & x_{2R} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NR} \end{bmatrix}_{N\times R},$$

$$B(q) = \begin{bmatrix} b_{11q} & b_{12q} & \cdots & b_{1Lq} \\ b_{21q} & b_{22q} & \cdots & b_{2Lq} \\ \vdots & \vdots & \vdots & \vdots \\ b_{R1q} & b_{R2q} & \cdots & b_{RLq} \end{bmatrix}_{R\times L}, \quad \Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_R),$$

$$[\mathbf{W}_{J\times L}]_{j,l} = \int K(m_j, m)K(m_l, m)dm,$$

$$[\mathbf{W}_{L\times L}]_{l_1,l_2} = \int K(m_{l_1}, m)K(m_{l_2}, m)dm,$$

$$\mathbf{K}_{l_1,l_2} = K(m_{l_1}, m_{l_2}).$$

The objective function (3.2.4) becomes to find $B(q)$ that minimizes

$$\text{trace}\{-2A(q)\mathbf{W}_{J\times L}B(q)^\top X^\top + XB(q)\mathbf{W}_{L\times L}B(q)^\top X^\top\} + \text{trace}\{\Lambda B(q)\mathbf{K}B(q)^\top\}.$$

Differentiate with regards to $B(q)$ would yield:

$$-2X^\top A(q)\mathbf{W}_{J\times L} + 2X^\top XB(q)\mathbf{W}_{L\times L} + 2\Lambda B(q)\mathbf{K} = 0.$$

Divide by 2 and vectorize the whole thing:

$$(\mathbf{W}_{L\times L} \otimes (X^\top X))vec(B(q)) - vec(X^\top A(q)\mathbf{W}_{J\times L}) + (\mathbf{K} \otimes \Lambda)vec(B(q)) = 0.$$

Therefore, the least square estimate of $B(q)$ can be found by:

$$vec(\hat{B}(q)) = (\mathbf{W}_{L\times L} \otimes (X^\top X) + \mathbf{K} \otimes \Lambda)^{-1} vec(X^\top A(q)\mathbf{W}_{J\times L}). \tag{3.2.5}$$

3.3 Computation

The algorithms presented are very high-dimension and it is not easily tractable. For example, with ADAPT data we have 7150 measurements per face, so in order to solve for (3.2.2), we now have to take an inverse of (7150·7150)×(7150·7150) matrix. In order to make this computation feasible, we intoduce using the eigenfunctions of RKHS as basis functions. This means that we use v_j that represent kernel as

$$K(u, u') = \sum_{j=1}^{\infty} \tau_j v_j(u) v_j(u'),$$

where τ_j are eigenvalues. The approximation of these eigenfunctions are discussed earlier in Section 1.1.2.2.

With v_j which are orthonormal in L^2 and orthogonal in $\mathbb{K}$ with norms of $1/\tau_j$, $\mathbf{W}$ becomes an identity matrix and K becomes a diagonal matrix with its diagonal terms being $1/\tau_j$. This setup allows us to find β using the algorithms presented in Section 3.2.

3.4 Comparison

In order to compare the prediction performance of the three algorithms, we take 10-fold cross validation and check the prediction errors. The covariates we took are the same as in Section 2.3.3: sex, age, height, weight, the population proportions from East Asian, South Asian, Southern European, Native American, and West

African. The relative prediction error is calculated as

$$\text{Relative Prediction Error:} \qquad \frac{SSE_{mean} - SSE_{reg}}{SSE_{mean}}$$

where SSE_{mean} is the sum of squared prediction errors when the model is only with the coefficient function (i.e. mean estimation) whereas SSE_{reg} is the sum of squared prediction errors with the full model. This is similar to R^2 and reveals relatively how much errors are explained by the predictors in the model.

	PCR	smoothPCR $\lambda = 10^{-15}$	smoothPCR $\lambda = 10^{-20}$	penal reg
RPE	0.2944	0.2940	0.2944	0.1568
Computation Time	30min	2hr	2hr	10min

Table 3.1: Comparison between three algorithms using the mean relative prediction error of 10-fold cross validation and computation time.

We took 99% PC level for both PCR and smooth PCR. The resulting mean prediction error for PCR is 0.2944 with number of principal components being about 100. The mean prediction error for smooth PCR depends on the smoothing parameter. With the smoothing parameter of 10^{-20}, it gives the mean relative prediction error of 0.2944, same as PCR case. The number of principal components is also very similar to be about 100. With the smoothing parameter of 10^{-15}, the model gives the mean relative prediction error of 0.2940, which is smaller but still very close to that of 10^{-20}. The mean prediction error for penalized regression is 0.1568, which is much less than the previous two. This means that penalized regression does the worst in terms of prediction. This is rather surprising because we believe that the smoothing on beta would increase the prediction power. But it may be because for penalized regression model, we predict each coordinate separately, ignoring the dependencies between the three coordinates. This also

emphasizes the importance of considering the three coordinates together when we conduct an analysis on facial shape.

In terms of the computation time, each cross validation took about 30 minutes for PCR, 2 hours for smooth PCR, and 10 minutes for penalized regression. The time consumption of smooth PCR comes largely from the spectral decomposition of the estimated covariance operator. Because the dimension of this covariance operator is $J\times3\times J\times3$ where J corresponds to the number of kernel eigenfunctions, and for our case, J is 4002. The same computation is required for PCR too, but we did a trick of conducting spectral decomposition on the kernel coefficients, without constructing the covariance operator. When the kernel coefficients form a matrix A, then the estimated covariance operator will be in the form of $C = \frac{A^\top A}{n-1}$ (assuming A is centered). This means that if we get spectral decomposition of A such that $A = USV^\top$, then $C = \frac{VSU^\top USV^\top}{n-1} = V\frac{S^2}{n-1}V^\top$. Thus we are able to get the spectral decomposition on C by getting the spectral decomposition on A. Since A in our case is of the dimension $N \times J \times 3$, conducting spectral decomposition on this instead of C which is of dimension $J\times3\times J\times3$ reduces down the computation time notably.

We apply the three algorithms to the whole ADAPT data and get the estimated beta functions. When we check the directional plots as in Figure 3.2, the beta estimated through PCR and smoothPCR are very similar although they are somewhat different from the beta estimated using felspline as in Section 2.3.3. However, the most notable difference comes from the beta estimates through the penalized regression. Since penalized regression is the only method that has not gone through the principal component analysis and the three coordinates are considered separately, that may account for why the predictors in penalized regression affect the response in somewhat different way from the PCR methods.

However, the beta estimates from penalized regression still affect the faces in a similar fashion. For sex effect, the cheek is rounder for the females, and that has been captured on the top right plot of Figure 3.2. For height effect, a taller person will have slender cheek if all the other predictors remain the same, and that is also captured as blue on the cheek for the beta from penalized regression. But for the population proportion from Western African, the tear-looking blue dots on the cheek disappear on the penalized regression, but this disappearance may be due to the smoothing on beta.

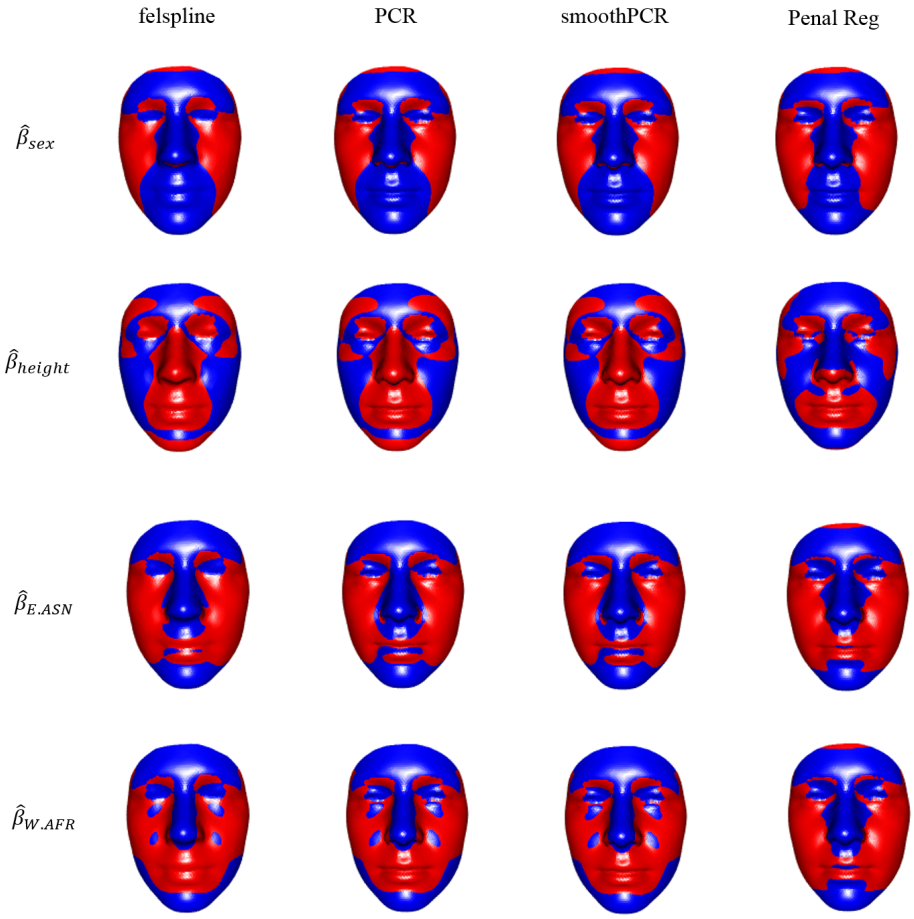

Figure 3.2: Comparison between the estimated beta using directional plots. Red means outward effect and blue means inward effect.

Chapter 4

Optimal Function-on-Scalar Regression over Complex Domains

4.1 Introduction

In this Chapter we establish the optimality of parameter estimation for function-on-scalar regression over complex domains by (1) establishing minimax lower bounds on the estimation rate and (2) providing a minimax optimal estimator whose upper bounds match the developed lower bounds.

We develop our theory under a fairly general structure:

$$Y_{ij\ell} = Y_{i\ell}(u_{ij}) + \delta_{ij\ell} = \sum_{k=1}^{K} X_{ik}\beta_{\ell k}(u_{ij}) + \varepsilon_{ij\ell}(u_{ij}) + \delta_{ij\ell}. \tag{4.1.1}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, m_i$, and $\ell = 1, \ldots, L$. Here i indexes the subject, j the observed domain point, and ℓ the coordinates of the functional outcomes. Intuitively, this means that for each subject we have a L functional outcomes $Y_{i\ell}(u) \in \mathbb{R}$ that are only observed at points $u_{ij} \in \mathcal{U}$. The domain $\mathcal{U}$ is most commonly the interval $[0, 1]$, but it may also be a more complex manifold, both of

which are included in our theory. Lila and Aston (2017) consider the thickness of the internal carotid artery meaning that $\mathcal{U}$ is a two dimensional manifold sitting in 3D space and $L = 1$. In our FMDA framework as in Chapter 2, we consider the shape of human faces, so our framework results in $\mathcal{U}$ being a two dimensional manifold while $L = 3$ since the face is measured in 3D. The dimension of $\mathcal{U}$ plays a critical role in the minimax estimation rates for the $\beta_{\ell k}(u)$, while, interestingly, the value L does not. In addition, it was previously thought that, in simpler settings such as mean estimation, it was necessary to control the smoothness of the underlying functions $Y_{i\ell}(u)$, or equivalently the errors $\varepsilon_{ij}(t)$. However, we show that this is actually unnecessary and establish all of our results under the very mild assumption that $\mathrm{E}\,\|\varepsilon_{ij}\|^2_{L^2(\mathcal{U})} < \infty$.

Another aspect of our work is to establish our convergence rates more broadly than just making assumptions about derivatives. We only assume that the $\beta_{\ell k}$ lie in an RKHS, and establish our rates relative to the rate of decay of the eigenvalues of the kernel defining the RKHS. Under mild assumptions, we will show that the optimal rate of converge is given by

$$O_P\left((nm)^{-\frac{2h}{2h+1}} + n^{-1}\right),$$

where h is connected to the kernel of the RKHS. When the dimension of $\mathcal{U}$ is d and the parameters $\beta_{\ell k}$ possess p derivatives, it can be shown that $h = p/d$ resulting in the rate

$$O_P\left((nm)^{-\frac{2p}{2p+d}} + n^{-1}\right),$$

thus we can clearly see how the dimension of $\mathcal{U}$ affects the convergence rates of our estimators, with higher dimensions resulting in slower rates. This highlights why it is so useful to exploit manifold structures that reside in higher dimensional spaces;

the convergence rate is tied to the dimension of the manifold, not the ambient space.

4.2 Modeling Assumptions

We now state our modeling assumptions one at a time. We will summarize at the end of this Section with a quick reference of all of the assumptions made. We begin with the relationship

$$Y_{i\ell}(u) = \sum_{k=1}^{K} X_{ik}\beta_{\ell k}(u) + \varepsilon_{i\ell}(u).$$

This represents the model for the underlying trajectories, which are not completely observed. The parameters, $\beta_{\ell k}$ are assumed to lie within $\mathbb{K}$. Regularity assumptions about the $\beta_{\ell k}$ are introduced by making assumptions about $\mathbb{K}$, especially the rate at which the eigenvalues of K converge to zero.

We make only minimal assumptions about the regularity of the $\varepsilon_{i\ell}(u)$. In particular, we will establish our minimax rates under the mild assumption the point-wise variance of the errors is bounded, $\sup_{u\in\mathcal{U}} \mathrm{Var}(\varepsilon_i(u)) < \infty$, which implies (and is only slightly stronger than) $\mathrm{E}\,\|\varepsilon_{i\ell}\|^2 < \infty$. In Yuan and Cai (2010) the much stronger assumption was made that $\mathrm{E}\,\|\varepsilon_{i\ell}\|_{\mathbb{K}}^2 < \infty$, which, by the reproducing property implies our assumption. While seemingly innocent, this is an incredibly strong assumption that would actually preclude achieving optimal convergence rates in most settings. Practically, the data is usually much rougher than the underlying mean parameters. However, Yuan and Cai (2010) requires that they reside in the same space, meaning that the $\beta_{\ell k}$ could only be smoothed up to the smoothness of the data. For example, if $\mathcal{U} = [0, 1]$ and $\beta_{k\ell}$ possessed two derivatives, while the $\varepsilon_{i\ell}$ only possessed one, then the rate given by Yuan and Cai

(2010) would be $(nm)^{2/3}+n^{-1}$, however, as we will show, this rate can be improved to $(nm)^{4/5}+n^{-1}$. Furthermore, in settings such as finance or the geosciences, the $\varepsilon_{i\ell}$ might not possess any derivatives or be part of any RKHS (e.g. Brownian motion).

Lastly, we will treat the X_{ij} as random variables with finite variance that are independent of the $\varepsilon_{ij\ell}$. The observed points u_{ij} will be assumed to be iid draws from $\mathcal{U}$, with density $f(u)$ that is bounded away from 0 and ∞. Note that since $\mathcal{U}$ is a manifold, the density $f(u)$ is with respect to Lebesgue measure over $\mathcal{U}$. We also assume that $Y_{i\ell}$ is observed with error, namely $Y_{ij\ell} = Y_{i\ell}(u_{ij}) = Y_{i\ell}(u_{ij}) + \delta_{ij\ell}$. The error $\delta_{ij\ell}$ are assumed to be iid across i and j, though they can be dependent in ℓ. We assume these errors are centered and have finite variance. We now summarize all of the assumptions introduced in this Section.

Assumption 4.2.1. *We make the following modeling assumptions.*

1. *The observed data are* $\{Y_{ij\ell}, X_{i1}, \ldots, X_{iK}\}$ *for* $i = 1, \ldots, n$, $j = 1, \ldots, m_i$, *and* $\ell = 1, \ldots, L$.

2. *The observed data satisfy the linear model*

$$Y_{ij\ell}(u_{ij}) = \sum_{k=1}^{K} X_{ik}\beta_{\ell k}(u_{ij}) + \varepsilon_{i\ell}(u_{ij}) + \delta_{ij\ell},$$

   where $u \in \mathcal{U} \subset \mathbb{R}^D$ *is a compact* d*-dimensional manifold with* $d \leq D$.

3. *The mean parameters reside within the RKHS,* $\beta_{\ell k} \in \mathbb{K}$, *with continuous kernel* $K(u, u')$.

4. *The sequences* $X_{ik} \in \mathbb{R}$, $\varepsilon_{i\ell} \in L^2$, $u_{ij} \in \mathcal{U}$, *and* $\delta_{ij\ell}$ *are random and independent of each other.*

5. *The covariates X_{ik} are potentially dependent across k, but iid across i. The vector $X_i = \{X_{ik}\}$ is assumed to have a finite covariance matrix with full rank.*

6. *The $\delta_{ij\ell}$ represent measurement error and are iid across i and j, though potentially dependent across ℓ. They have mean zero and finite variance.*

7. *The stochastic processes $\varepsilon_{i\ell}$ are iid across i, though potentially dependent across ℓ. They are assumed to have mean zero and to satisfy $\sup_{u\in\mathcal{U}} \mathrm{Var}(\varepsilon_{i\ell}(u)) < \infty$.*

4.3 Theoretical Results

We now provide three key theoretical results. The first is a lower bound on the best possible estimation rate. This bound is obtained using an application of Fano's lemma. Second, we provide an estimator whose upper bound matches the lower bound, implying that it is optimal in a minimax sense. Lastly, under slightly stronger assumptions, we show that our estimator converges in distribution, which will allow practitioners to carry out statistical inference on the $\beta_{k\ell}$.

4.3.1 Lower Bound

Recall that when referring to a minimax rate, have to specify the loss function as well as the class of models we are considering. In this case, our loss will be $L^2(\mathcal{U})$, and we consider all models as outline in Assumption 4.2.1. In the minimax rate, the distributions of the X_{ij}, $\varepsilon_{i\ell}$, and $\delta_{ij\ell}$ are fixed, and thus the models will be indexed by the $\beta_{\ell k}$, which we assume lie in a closed bounded ball of $\mathbb{K}$: $\|\beta_{\ell k}\|_{\mathbb{K}} \leq M_0$, which will be denoted as $B_{\mathbb{K}}$.

Before giving the minimax rate, it is useful to first define the excess risk:

$$R_n = \sum_{k=1}^{K} \sum_{\ell=1}^{L} \|\hat{\beta}_{\ell k} - \beta_{\ell k}\|^2.$$

We say that the rate of convergence of $\hat{\beta}$ is a_n if $R_n = O_P(a_n)$. The minimax estimation risk is then defined as the optimal rate of convergence (i.e. the smallest a_n) of the worst case modeling scenario. More precisely, we say the minimax rate is a_n if

$$\liminf_{n\to\infty} \min_{\{\hat{\beta}_{\ell k}\}} \max_{\{\beta_{\ell k} \in B_{\mathbb{K}}\}} P(a_n \epsilon \leq R_n \leq a_n \epsilon^{-1}) \to 1 \qquad \text{as } \epsilon \to 0.$$

The left hand side of the inequality is the lower bound, which indicates the lower bound on the risk of the best possible estimator.

Theorem 4.3.1. *Under Assumption 4.2.1 the excess risk satisfies*

$$\liminf_{n\to\infty} \min_{\{\hat{\beta}_{\ell k}\}} \max_{\{\beta_{\ell k} \in B_{\mathbb{K}}\}} P(R_n \geq \epsilon((nm)^{-2h/(2h+1)} + n^{-1})) \to 1 \qquad \textit{as } \epsilon \to 0,$$

where the estimates $\{\hat{\beta}_{\ell k}\}$ are functions of the observed data.

The proof of Theorem 4.3.1 is given in the appendix. It shows that no estimator can achieve a rate faster than $(nm)^{-2h/(2h+1)} + n^{-1}$; we will show in the next Section that is is bound is tight by giving an estimator that achieves the lower bound. The proof is based on an application of Fano's lemma. We show that a sequence of parameters within the ball $B_{\mathbb{K}}$ can be selected which are sufficiently far apart with respect to the $\mathbb{K}$ norm. We then prove a bound the Kullback-Leibler divergence between any pair of probability measures induced by this collection of parameters. Combining these two bounds, we are able to apply Fano's lemma to obtain the desired result.

4.3.2 Upper Bound

To prove the upper bound, we need only construct an estimator that achieves the lower bound. We will then know that the lower bound is tight and that the selected estimator is minimax. The estimator we propose here is a slight variant of the one used in Section 4.4.2, however it makes the mathematical arguments clearer and also shows how the minimax rate for function-on-scalar regression is intimately tied to the minimax rate for mean estimation.

We construct our estimator in two parts by noticing that

$$\boldsymbol{\beta}^\top(u) = \mathbf{C}_X^{-1}\mathbf{C}_{XY}(u),$$

where $\mathbf{C}_X = \mathrm{E}[X_i X_i^\top]$ and $\mathbf{C}_{XY} = \mathrm{E}[X_i Y_i^\top(u)]$. Notice $\boldsymbol{\beta}^\top(u)$ is a $K \times L$ matrix. The first term is estimated using

$$\widehat{\mathbf{C}}_X = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^\top.$$

By Assumption 4.2.1, $\widehat{\mathbf{C}}_X = \mathbf{C}_X + O_P(n^{-1/2})$ where $\mathbf{C}_X = \mathrm{E}[X_i^\top X_i]$, and since $\mathbf{C}_X$ has full rank $\widehat{\mathbf{C}}_X^{-1} = \mathbf{C}_X^{-1} + O_P(n^{-1/2})$ for n large.

The second quantity we estimate coordinate-wise. Notice that $\mathbf{C}_{XY}(u)$ is a $K \times L$ dimensional matrix of functions and by Assumption 4.2.1 each coordinate is in $\mathbb{K}$. As we will show in the Section 4.5, each coordinate can be estimated at the rate $(nm)^{-2h/(2h+1)} + n^{-1}$, which combined with Slutsky's lemma gives the desired result.

Theorem 4.3.2. *Assume that 4.2.1 holds and that* $\hat{\boldsymbol{\beta}}^\top(u) = \hat{\mathbf{C}}_X^{-1}\hat{\mathbf{C}}_{XY}(u)$ *as*

described above. Then the excess risk satisfies

$$\liminf_{n\to\infty} \min_{\{\hat{\beta}_{\ell k}\}} \max_{\{\beta_{\ell k \in B_{\mathbb{K}}}\}} P(R_n \leq \epsilon^{-1}((nm)^{-2h/(2h+1)} + n^{-1})) \to 1 \qquad as\ \epsilon \to 0.$$

Combining Theorems 4.3.1 and 4.3.2 we get that the minimax rate of converges is $(nm)^{-2h/(2h+1)} + n^{-1}$. Furthermore, this rate holds quite broadly across different $\mathbb{K}$. We now discuss this rate in a bit more detail.

The *phase-transition* occurs when the rate becomes parametric, i.e., n^{-1}. Clearly this occurs if

$$(nm)^{-2h/(2h+1)} \ll n^{-1} \Longrightarrow m \gg n^{1/2h}.$$

In other words, the rate becomes parametric if the (harmonic) average number of points per curve is more than $n^{1/h}$. If m is less, then the rate is slower than parametric. In the worst case, when m is bounded, the rate becomes the classic nonparametric rate $n^{-h/(h+1)}$.

Another major point concerns of the value of h. Currently, h has only been tied to the rate of decay of the eigenvalues of the RKHS kernel. However, there are a few settings where this rate can be made more interpretable. In particular, Sobolev spaces with inner product norms are RKHS when certain kernels are taken, like the Matérn kernel. If the functions possess p derivatives and the dimension of $\mathcal{U}$ is d, then we have $h = p/d$, and thus the minimax rates become

$$(nm)^{\frac{-2p}{2p+d}} + n^{-1}.$$

We can thus see the effect of the dimension of the domain on the rates. As we move to higher dimensions the rates get worse, while they improve if the parameters have

more derivatives.

4.4 Numerical Illustrations

In this Section we provide simulations in the case where the outcome is one dimensional and there is one predictor:

$$Y_i(u) = X_i\beta(u) + \varepsilon_i(u).$$

We use this simplified setting to illustrate the effect of the dimension of the domain on the estimation rates of β. We first present our simulation setting, briefly describe how our estimators are computed, and then show the simulation results.

4.4.1 Simulation Setting

In this Section we show the convergence of the estimation error of β_{ik} using simulations. There are two simulation settings taken: (1) $\mathcal{U} = [0,1] \subset \mathbb{R}$ case, meaning that the domain is one-dimensional, and (2) $\mathcal{U} = [0,1] \times [0,1] \subset \mathbb{R}^2$ case, meaning that the domain is two-dimensional.

The construction of $Y_{ij\ell}(u_{ij})$ is as

$$Y_{ij\ell}(u_{ij}) = X_i\beta_\ell(u_{ij}) + \varepsilon_{i\ell}(u_{ij}) + \delta_{ij\ell},$$

where the covariate X_i are taken as iid $\mathcal{N}(10,1)$ and the measurement errors taken as iid $\mathcal{N}(0,0.05)$.

The construction of $\beta_\ell(u_{ij})$ is done using Fourier series. The degree of smoothness of a function is implied by the rate of decay of its Fourier series coefficients. If the decay is faster, the function would be smoother. When a

function $f(x)$ is $C^n[0,1]$ for some $n > 0$, meaning that its n^{th} derivative is continuous, then the fourier series coefficients $a_k = O\left(\frac{1}{k^{n+1}}\right)$. Therefore, we can implicitly control the number of continuous derivatives of β_ℓ by choosing the power of k. For one-dimensional $\mathcal{U}$, β_ℓ is constructed as

$$\beta_\ell^{(1)}(u) = \sum_{k=1}^{100} \frac{1}{\pi k^{\alpha^{(1)}}} cos(k\pi u)$$

and for two-dimensional $\mathcal{U}$, let $u = (u_1, u_2)$,

$$\beta_\ell^{(2)}(u_1, u_2) = \sum_{k=1}^{100}\sum_{r=1}^{100} \frac{1}{\pi (kr)^{\alpha^{(2)}}} cos(k\pi u_1) cos(r\pi u_2).$$

The superscripts (1) and (2) of β_ℓ and α stands for the one dimension case and the two dimension case. This construction lets us to choose the number of continuous derivatives of β_ℓ by choosing α since in each case the number of derivatives is implied to be $\alpha - 1$. Since we would like to set the rate of decay of the eigenvalues of RKHS kernel (if the resulting RKHS is equivalent to a Sobolev space) h is considered as p/d where p is the number of derivatives the function possess and d is the dimension of the data.

Therefore, we take $\alpha^{(1)} = 3, 4, 5$ for the one-dimensional case, which implies $h = 2, 3, 4$, and we take $\alpha^{(2)} = 5, 7, 9$ for the two-dimensional case, which also implies $h = 2, 3, 4$. The corresponding β's are visualized in Figure 4.1 and Figure 4.2. The change of α does not seem to make obvious change in the shape of the resulting β functions, but with higher α, the β function is smoother.

The error function $\varepsilon_{i\ell}(u)$ is taken as the linear combination of orthonormal cosine bases, such that

$$\varepsilon_{i\ell}^{(1)}(u) = \sum_{k=1}^{10} e_k^{(1)} cos(k\pi u)$$

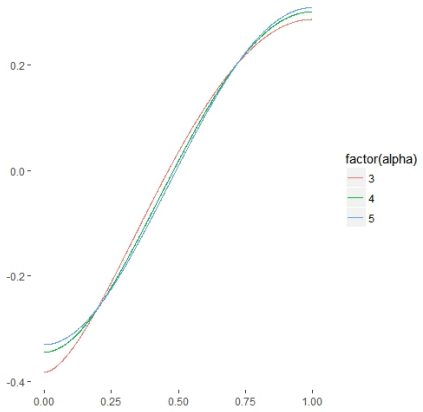

Figure 4.1: The visualization of $\beta^{(1)}$ corresponding to $\alpha^{(1)} = 3, 4, 5$.

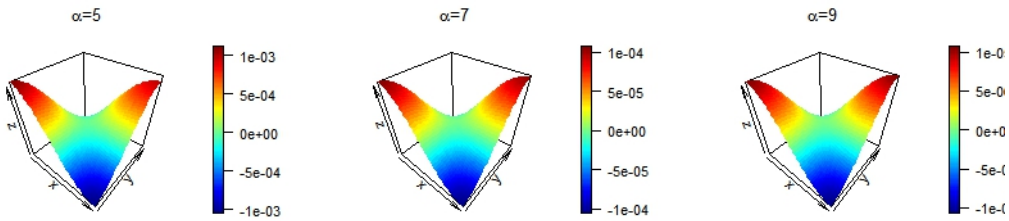

Figure 4.2: The visualization of $\beta^{(2)}$ corresponding to $\alpha^{(1)} = 5, 7, 9$.

for the one-dimensional case, and

$$\varepsilon_{i\ell}^{(2)}(u_1, u_2) = \sum_{k=1}^{5} \sum_{r=1}^{5} e_{kr}^{(2)} cos(k\pi u_1) cos(k\pi u_2)$$

where $e_k^{(1)}$ and $e_{kr}^{(2)}$ are generated from $\mathcal{N}(0, 0.01)$.

We use Matérn kernel for the RKHS space since using Matérn kernel, we can have the resulting RKHS equivalent to a Sobolev space. Matérn also allows us to choose the smoothness parameter ν which dictates the smoothness of the space.

The Matérn kernel has the form

$$K_\nu(u,u') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|u-u'\|}{\rho}\right) K_\nu \left(\sqrt{2\nu}\frac{\|u-u'\|}{\rho}\right),$$

where $\|u-u'\|$ is the Euclidean distance between two points inputted. We have chosen $\sigma = 1$ and $\rho = 5$ and try different ν's like $\nu = 1/2$, $\nu = 3/2$, $\nu = 5/2$. When $\nu = 1/2$, the kernel becomes equivalent to an exponential kernel.

The results of these simulation settings will be discussed after we talk about the computation.

4.4.2 Computation

Using the representer theorem one can obtain an exact expression for the estimator. However, this turns out to be very inefficient computationally as it involves solving for $\sum_i m_i$ parameters. Instead, we will express the estimator using the first R eigenfunctions of $K(u,u')$:

$$\beta_R(u) = \sum_{r=1}^{R} b_r v_r(u).$$

We provide an exact form for the the coefficients $\{b_{jr}\}$. As long as R is chosen large enough, then the truncation error will be of a lower order than the convergence rate. If the β all like in a $\mathbb{K}$ ball then the truncation error is of the order

$$\|\beta_0 - \beta_R\|^2 = \sum_{r=R+1}^{\infty} b_r^2 = \sum_{r=R+1}^{\infty} \tau_r \frac{b_r^2}{\tau_r} \leq \tau_R \|\beta_0\|_{\mathbb{K}}^2 \asymp R^{-h}.$$

Thus we see that as $R \gg n^{1/h}$ and $R \gg (nm)^{1/(h+1)}$ then the truncation error will be asymptotically negligible. Of course, in practice, one can take R much larger as long as the computational resources allow.

For simplicity, we assume that $m_i \equiv m$, but the general case can be handled by reweighting the X_{ik} and Y_{ij} and using $\bar{m}$ in place of m. The target function is now given by

$$\ell_{nm,\lambda}(\mathbf{b}) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\left(Y_{ij} - \sum_{k=1}^{K}\sum_{r=1}^{R} X_{ik}b_{kr}v_r(u_{ij})\right)^2 + \lambda\sum_{k=1}^{K}\sum_{r=1}^{R}\frac{b_{kr}^2}{\tau_k}.$$

We will rewrite this expression using vector/matrix notation. First, let $b_v = vec(\mathbf{b})$, where vec denote stacking the columns into a single vector. Properties of the vec operation imply that

$$\sum_{r=1}^{R} X_{ik}b_{kr}v_r(u_{ij}) = X_i^\top \mathbf{b} V_{ij} = (V_{ij}^\top \otimes X_i^\top)b_v.$$

Define

$$Y_v = \mathrm{vec}(\mathbf{Y}) \qquad \mathbf{A} = \begin{pmatrix} (V_{11}^\top \otimes X_1^\top) \\ (V_{21}^\top \otimes X_2^\top) \\ \vdots \\ (V_{m1}^\top \otimes X_m^\top) \\ (V_{21}^\top \otimes X_1^\top) \\ (V_{22}^\top \otimes X_2^\top) \\ \vdots \end{pmatrix} \qquad \mathrm{T} = \begin{pmatrix} \tau_1 & 0 & \dots & 0 \\ 0 & \tau_2 & \dots & 0 \\ \vdots & \ddots & \dots & 0 \\ 0 & \dots & 0 & \tau_R \end{pmatrix}$$

Then the target function becomes

$$\frac{1}{nm}(Y_v - \mathbf{A}b_v)^\top(Y_v - \mathbf{A}b_v) + \lambda b_v^\top(\mathrm{T}^{-1} \otimes I)b_v.$$

The solution can then be expressed as

$$\hat{b}_v = \left((nm)^{-1}\mathbf{A}^\top\mathbf{A} + \lambda(\mathrm{T}^{-1} \otimes I)\right)^{-1} (nm)^{-1}\mathbf{A}^\top Y_v.$$

4.4.3 Simulation Results

The simulation results are shown in Figure 4.3 and Figure 4.4. The y-axis stands for the median estimation error based on 500 simulation runs while the x-axis stands for the sample size n for Figure 4.3 and the number of measurements per curve m for Figure 4.4.

In both Figures, the top three plots correspond to one-dimensional cases while the bottom three correspond to the two-dimensional cases. The columns represent the choice of ν in Matérn kernel; the left plots are with $\nu = 1/2$, the middle plots are with $\nu = 3/2$, and the right plots are with $\nu = 5/2$. The color of lines represents the size of m in Figure 4.3 and the size of n in Figure 4.4. The larger points on the plots correspond to higher α values.

As expected, the estimation error decreases exponentially with increasing n, as shown in Figure 4.3. Notice that we have plotted n up to 50 for the one-dimensional cases on the top whereas we have plotted n up to 200 for the two-dimensional cases. This implies that the convergence happens much slower with the higher dimension of the domain. The estimation errors are mostly larger with smaller α, as the lower α means that the β is rougher, except for the top middle and top right plots where the order is reversed; the estimation error is highest with the largest α, meaning the smoothest β, but this is when both n and m are very small ($n = 5$ and $m = 5$). If we take only 5 points from 1,000 grid on the domain, and n is also very small, this combination just cannot estimate the true β well. Except for those cases, the estimation errors turned out to be as we expected.

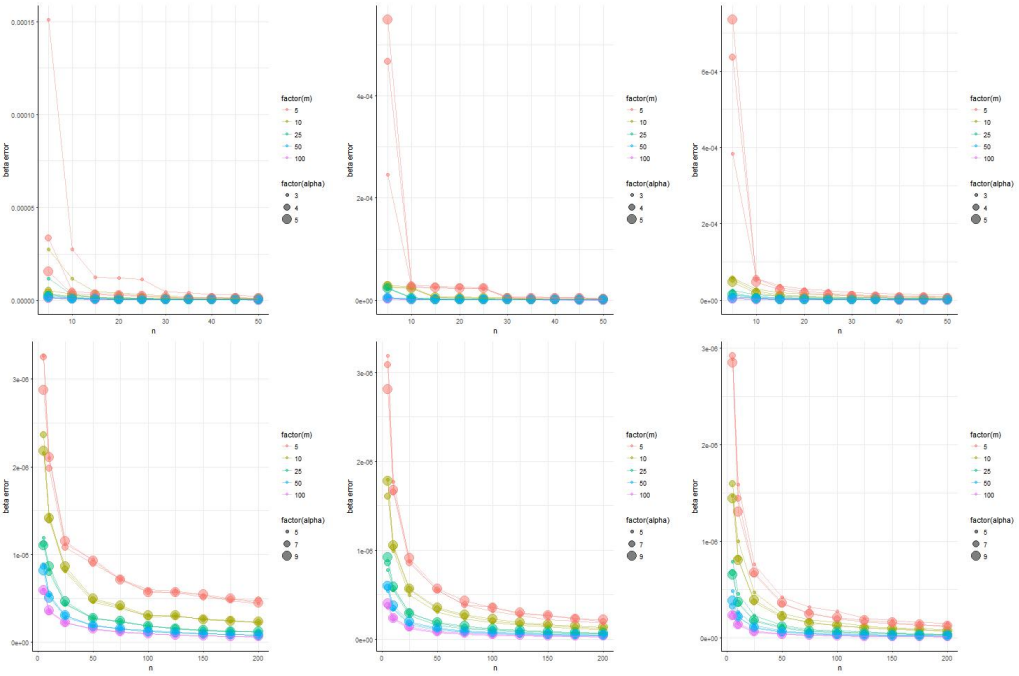

Figure 4.3: The plots show how the estimaton errors are affected by the number of samples (n). The 1d cases are on the top and the 2d cases are on the bottom. From left to right, the smoothness parameter ν of Matérn kernel is taken as 1/2, 3/2, 5/2.

With increasing m also, the estimation error decreases exponentially, as shown in Figure 4.4. Here, for all one-dimensional cases on the top and two-dimensional cases on the bottom, m is plotted up to 200. We can see here also that the convergence of the estimation error happens much faster with one-dimensional cases. It is also good to note that for large n, like $n = 100$ shown with the purple color in the plot, the estimation error does not seem to change after certain level of m. This shows that the n^{-1} term dominating $(nm)^{\frac{-2h}{2h+1}}$ term, and m does not affect the estimation error as much.

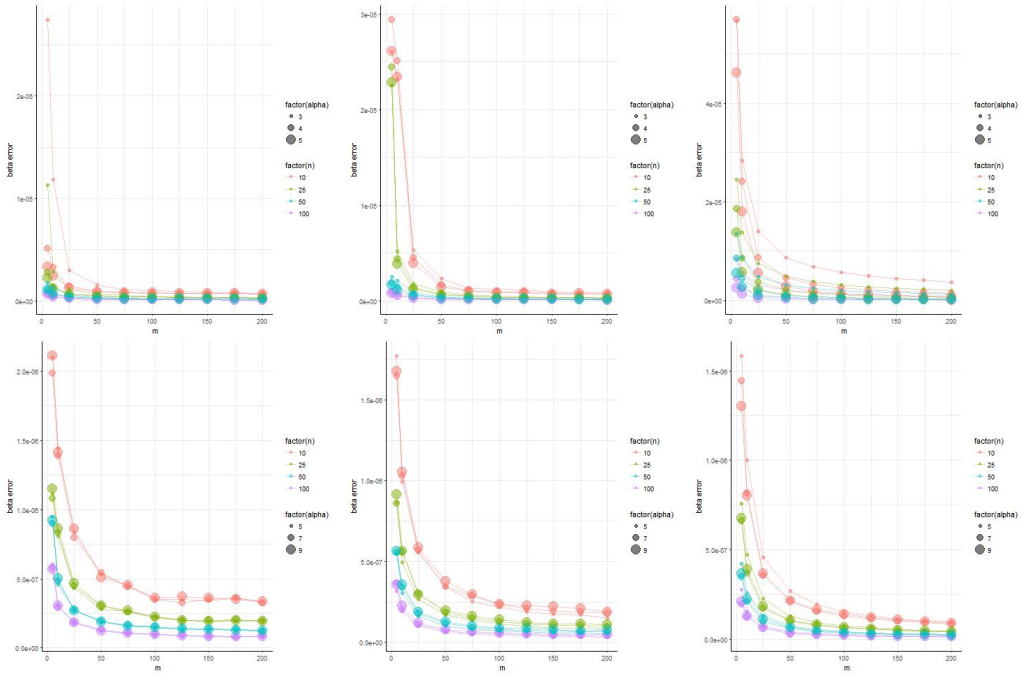

Figure 4.4: The plots show how the estimaton errors are affected by the number of points per curve (m). The 1d cases are on the top and the 2d cases are on the bottom. From left to right, the smoothness parameter ν of Matérn kernel is taken as 1/2, 3/2, 5/2.

4.5 Technical Proofs

4.5.1 Proof of lower bound

Let $m_i \equiv m$ and define $N = c(nm)^{1/(1+2r)}$. Let $b = (b_1, \ldots, b_N)$ with $b_i \in \{0, 1\}$. Now define the functions, for some fixed M_0,

$$g_b(t) = M_0^{1/2} N^{-1/2} \sum_{k=N+1}^{2N} \tau_k^{1/2} b_{k-N} v_k(t).$$

Then we have that the $\mathbb{K}$ norm is given by

$$\|g_b\|_{\mathbb{K}}^2 = M_0 N^{-1} \sum_{k=N+1}^{2N} \tau_k b_{k-N} \tau_k^{-1} \leq M_0.$$

If b' is another sequence in $\{0,1\}^N$ then

$$\begin{aligned}\|g_b - g_{b'}\|^2 &= M_0 N^{-1} \sum_{k=N+1}^{2N} \tau_k (b_{k-N} - b'_{k-N})^2 \\ &\geq M_0 N^{-1} \tau_{2N} \sum_{k=N+1}^{2N} (b_{k-N} - b'_{k-N})^2 \\ &\geq c_0 N^{-1} N^{-2r} H(b, b') = c_0 N^{-(1+2r)} H(b, b'),\end{aligned}$$

where c_0 is some constant. The Varshamov-Gilbert bound implies there exists binary sequences $b^{(i)} \in \{0,1\}^N$ for $i = 1, \ldots, M$, with $M \geq 2^N$ that satisfy

$$H(b^{(i)}, b^{(j)}) \geq N \qquad i \neq j.$$

In which case we have

$$\|g_{b^{(i)}} - g_{b^{(j)}}\|^2 \geq c_0 N^{-2r}.$$

We can also get a similar upper bound on the difference by noting that $H(b, b') \leq N$ which then implies

$$\|g_b - g_{b'}\|^2 \leq M_0 N^{-1} \tau_{N+1} H(b, b') \leq C_0 N^{-1} N^{-2r} N = C_0 N^{-2r}.$$

Now consider the probability measure P_i over $\mathbb{R}^m$ which, conditioned on $T = (t_1, \ldots, t_m)^\top$, is multivariate normal with mean vector $\mu_i(T) := (g_{b^{(i)}}(t_1), \ldots, g_{b^{(i)}}(t_m))^\top$ and covariance matrix $\Sigma(T) := \{C(t_j, t_k) + \sigma_0^2 1_{j=k}\}$. Assume that the t_i are iid uniform over $\mathcal{T}$. We then have that the KL divergence between P_i and P_j is given by

$$KL(P_i, P_j) = n \, \mathrm{E}[(\mu_i(T) - \mu_j(T))^2 \Sigma(T)^{-1} (\mu_i(T) - \mu_j(T))]$$

$$
\begin{aligned}
&\leq n\sigma_0^{-2} \sum_{k=1}^{m} \mathrm{E}[(g_{b^{(i)}}(t_k) - g_{b^{(j)}}(t_k)]^2 \\
&= nm\sigma_0^{-2} \|g_{b^{(i)}} - g_{b^{(j)}}\|^2 \\
&\leq c_0 nm N^{-1/2r}.
\end{aligned}
$$

We can now apply Fano's lemma to obtain

$$
\max_{1\leq j\leq M} \mathrm{E}_{g_{b^{(j)}}} \|\tilde{g}_\lambda - g_{b^{(j)}}\| \geq c_0 N^{-r} \left(1 - \frac{\log(c_1 nm\sigma_0^{-2} N^{-2r}) + \log(2)}{\log(M)}\right) \asymp (nm)^{-r/(2r+1)},
$$

for any estimator $\tilde{g}$, which gives the desired lower bound.

4.5.2 Proof of upper bound

Assumption 4.5.1. *We make the following assumptions.*

1. *Assume that*

$$
Y_{ij} = g_0(t_{ij}) + X_i(t_{ij}) + \varepsilon_{ij},
$$

   for $i = 1, \ldots, n$ and $j = 1, \ldots, m_i$. Let $m = (n^{-1} \sum m_i^{-1})^{-1}$ denote the harmonic mean of the m_i, Here we assume that $t_{ij} \in \mathcal{T} \subset \mathbb{R}^d$. The region $\mathcal{T}$ is assumed to be compact with positive Lebesgue measure. The random variables, t_{ij}, are assumed to have a density (wrt Lebesgue measure), which is zero off of $\mathcal{T}$ and bounded above and below (from 0) on $\mathcal{T}$. Without loss of generality we will assume that $\mathcal{T}$ has Lebesgue measure 1 and that the t_{ij} are drawn uniformly, so t_{ij} has density 1.

2. *Let the X_i be iid mean zero elements of $L^2[0,1]$ with covariance function $C(t,s)$.*

3. *The covariance function $C(t,s)$ satisfies $\sup_{t\in\mathcal{T}} C(t,t) < \infty$, which implies $\mathrm{E}\,\|X_1\|^2 < \infty$.*

4. *The errors* ε_{ij} *are iid mean zero with* $0 < \mathrm{Var}(\varepsilon_{11}) < \infty$.

5. *The function* $K(t,s)$ *is symmetric, positive definite, and continuous over* $\mathcal{T} \times \mathcal{T}$. *We let* $\mathbb{K}$ *denote the RKHS with kernel* $K(t,s)$.

6. *The mean function* g_0 *satisfies* $\|g_0\|_{\mathbb{K}} < \infty$.

7. *The eigenvalues,* τ_k, *of* $K(t,s)$ *are of the order* $\tau_k \asymp k^{-2r}$, *for some* $r > 1$, *where* $\asymp$ *denotes that the ratio is bounded below from zero and above from* ∞.

We will use an RKHS framework for estimating $g_0(t)$. We assume the kernel $K(t,s)$ is continuous over $\mathcal{T}$, which means it is also bounded. Using Mercer's theorem it admits the spectral decomposition

$$K(t,s) = \sum_{i=1}^{\infty} \tau_i v_i(t) v_i(s). \tag{4.5.1}$$

Recall that, by Mercer's theorem, the convergence above occurs uniformly and absolutely in t and s. We therefore have the following lemma, which will be used throughout.

Lemma 4.5.1. *If* $K(t,s)$ *is a continuous, positive definite, and symmetric kernel then it admits the eigen-decomposition* (4.5.1), *which satisfies*

$$\sup_{t,s} \tau_k |v_k(t) v_k(s)| \to 0 \quad \text{as } k \to \infty.$$

The functions $v_k(t)$ are normalized to have $L^2(\mathcal{T})$ norm 1 (from here on we notationally drop the domain $\mathcal{T}$), which also means they have $\mathbb{K}$ norm τ_i^{-1}. Recall

that the $\mathbb{K}$ inner product can be expressed as

$$\langle g, f\rangle_{\mathbb{K}} = \sum_{i=1}^{\infty} \frac{\langle f, v_i\rangle\langle g, v_i\rangle}{\tau_i},$$

where norms and inner products without subscripts will always denote the L^2 norm. Define the target function as

$$\ell_{mn}(g) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}(Y_{ij} - g(t_{ij}))^2 + \lambda\|g\|_{\mathbb{K}}^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}(Y_{ij} - \langle g, K_{t_{ij}}\rangle_{\mathbb{K}})^2 + \lambda\|g\|_{\mathbb{K}}^2.$$

The minimizer, $\hat{g}$, can be obtained in a closed form using operator notation (as opposed to the representer theorem). We can take the derivative with respect to g (in the $\mathbb{K}$ topology) as

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} -2(Y_{ij} - \langle K_{t_{ij}}, g\rangle_{\mathbb{K}})K_{t_{ij}} + 2\lambda g,$$

where $K_{t_{ij}}(t) := K(t_{ij}, t)$. Define

$$h_{nm} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} Y_{ij}K_{t_{ij}}, \tag{4.5.2}$$

and the linear operator $T_{nm} : \mathbb{K} \to \mathbb{K}$ as

$$T_{nm}(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} f(t_{ij})K_{t_{ij}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\langle f, K_{t_{ij}}\rangle_{\mathbb{K}}K_{t_{ij}}. \tag{4.5.3}$$

Setting the derivative equal to zero we get the operator form for the estimator

$$-2h_{nm} + 2T_{mn}g + 2\lambda g = 0 \Longrightarrow \hat{g} = (T_{nm} + \lambda I)^{-1}h_{nm}.$$

We now define the biased population parameter that will act as an intermediate

value in our asymptotic derivation. Consider

$$\ell_{\infty,\lambda} = \mathrm{E}[(Y_{11} - g(t_{11}))^2] + \lambda \|g\|_{\mathbb{K}}^2.$$

Taking the derivative with respect to g we get

$$\frac{\partial}{\partial g}\ell_{\infty,\lambda} = -2\,\mathrm{E}[(Y_{11} - g(t_{11}))K_{t_{11}}] + 2\lambda g.$$

Define the linear operator $Tf := \mathrm{E}[f(t_{11})K_{t_{11}}] = \mathrm{E}[\langle K_{t_{11}}, f\rangle_{\mathbb{K}} K_{t_{11}}]$ and the transformed mean function $h = \mathrm{E}[Y_{11}K_{t_{11}}] = \mathrm{E}[g_0(t_{11})K_{t_{11}}] = Tg_0$. Notice that T can also be expressed as an integral operator with kernel $K(t,s)$:

$$[Tf](t) := \mathrm{E}[f(t_{11})K_{t_{11}}(t)] = \int K(t,s)f(s)\,ds.$$

We can set the derivative equal to zero to obtain

$$-2h + 2Tg + 2\lambda Ig = 0 \Longrightarrow g_\lambda = (T + \lambda I)^{-1}h = (T + \lambda I)^{-1}Tg_0. \tag{4.5.4}$$

We now define a final intermediate value as

$$\tilde{g}_\lambda = g_\lambda + (T + \lambda I)^{-1}(h_{nm} - T_{nm}(g_\lambda) - \lambda g_\lambda). \tag{4.5.5}$$

To establish our convergence rates we break up the problem into three pieces:

$$\hat{g} - g_0 = (g_\lambda - g_0) + (\tilde{g}_\lambda - g_\lambda) + (\hat{g} - \tilde{g}_\lambda).$$

In order to establish bounds for the third term above, it will be necessary to bound the second term in terms of the norm $\|f\|_\alpha = \langle K^{-\alpha/2}f, K^{-\alpha/2}f\rangle$. When $\alpha = 0$ this is the L^2 norm, when $\alpha = 1$ it is the $\mathbb{K}$ norm, but we allow intermediate values

$\alpha \in [0, 1]$.

Step 1: $g_\lambda - g_0$

Using (4.5.4) we have

$$g_\lambda - g_0 = [(T + \lambda)^{-1}T - I]g_0 = -\lambda(T + \lambda I)^{-1}g_0.$$

Notice that the eigenvalues of $(T + \lambda I)$ are $\tau_k + \lambda$ and the eigenfunctions are v_k Applying Parceval's identity yields

$$\begin{aligned}
\|g_\lambda - g_0\|^2 &= \lambda^2 \sum_k \frac{1}{(\tau_k + \lambda)^2} \langle g_0, v_k \rangle^2 \\
&= \lambda^2 \sum_k \frac{\tau_k}{(\tau_k + \lambda)^2} \frac{\langle g_0, v_k \rangle^2}{\tau_k} \\
&\leq \lambda^2 \|g_0\|_{\mathbb{K}}^2 \sup \frac{\tau_k}{(\tau_k + \lambda)^2}.
\end{aligned}$$

To bound the sup consider the function $f(x) = x^\gamma (x + \lambda)^{-2}$, over $x \geq 0$ and for some fixed $\gamma > 0$. Notice that this function will attain its maximum at a finite value of x if and only if $\gamma < 2$, for $\gamma \geq 2$ the maximum is attained at infinity. The derivative is given by

$$\gamma x^{\gamma - 1}(x + \lambda)^{-2} - 2x^\gamma (\lambda + x)^{-3}.$$

Setting equal to zero we have

$$\gamma(\lambda + x) - 2x = 0 \Longrightarrow x = \frac{\gamma}{2 - \gamma}\lambda.$$

So we have

$$\sup \frac{\tau_k^\gamma}{(\tau_k + \lambda)^2} \leq c_0 \lambda^{\gamma - 2}. \tag{4.5.6}$$

Note that throughout we take c_0, c_1, etc, to denote generic constants whose exact values may change depending on the context. Taking $\gamma = 1$ we conclude that

$$\|g_\lambda - g_0\|^2 \leq c_0 \lambda \|g_0\|^2. \tag{4.5.7}$$

Step 2: $\tilde{g}_\lambda - g_\lambda$

In this part we will bound the difference more generally using the α norm for $\alpha < 1 - 1/2r$. First, recall that, by definition of g_λ we have

$$Tg_\lambda + \lambda g_\lambda = h \Longrightarrow \lambda g_\lambda = h - Tg_\lambda = T(g_0 - g_\lambda).$$

Plugging this into (4.5.5), the expression for $\tilde{g}_\lambda$, we obtain

$$\tilde{g}_\lambda - g_\lambda = (T + \lambda I)^{-1} \left[h_{nm} - T_{nm} g_\lambda - (Tg_0 - Tg_\lambda). \right].$$

Using (4.5.2) notice that

$$\mathrm{E}[h_{nm}](t) = \mathrm{E}[Y_{11} K_{t_{11}}(t)] = (Tg_0)(t).$$

and similarly using (4.5.3)

$$\mathrm{E}[T_{nm} g_\lambda](t) = \mathrm{E}[f(t_{11}) K_{t_{11}}(t)] = (Tg_\lambda)(t),$$

Using Parceval's identity we get that the expected difference in the α norm is then given by

$$\mathrm{E}\,\|\tilde{g}_\lambda - g_\lambda\|_\alpha^2 = \sum_k \frac{1}{\tau_k^\alpha(\tau_k+\lambda)^2}\,\mathrm{Var}(\langle h_{nm} - T_{nm}g_\lambda, v_k\rangle).$$

Using the assumed independence across i and the definitions (4.5.2) and (4.5.3) we have

$$\mathrm{Var}(\langle h_{nm} - T_{nm}g_\lambda, v_k\rangle) = \frac{1}{n^2}\sum_i \frac{1}{m_i^2}\,\mathrm{Var}\left(\sum_j (Y_{ij} - g_\lambda(t_{ij}))\langle K_{t_{ij}}, v_k\rangle\right).$$

Using the representer theorem and that the v_k are the eigenfunctions of K, we can express $\langle K_{t_{ij}}, v_k\rangle = \tau_k\langle K_{t_{ij}}, v_k\rangle_{\mathbb{K}} = \tau_k v_k(t_{ij})$, so the above becomes

$$\frac{\tau_k^2}{n^2}\sum_i \frac{1}{m_i^2}\,\mathrm{Var}\left(\sum_j (Y_{ij} - g_\lambda(t_{ij}))v_k(t_{ij})\right).$$

Conditioning on the sigma algebra generated by the locations, $\mathcal{F} = \sigma\{t_{ij}\}$, we get

$$\begin{aligned}\mathrm{Var}\left(\sum_j (Y_{ij} - g_\lambda(t_{ij}))v_k(t_{ij})\right) &= \mathrm{Var}\left(\mathrm{E}\left[\sum_j (Y_{ij} - g_\lambda(t_{ij}))v_k(t_{ij})\middle|\mathcal{F}\right]\right)\\ &\quad + \mathrm{E}\left[\mathrm{Var}\left(\sum_j (Y_{ij} - g_\lambda(t_{ij}))v_k(t_{ij})\middle|\mathcal{F}\right)\right].\end{aligned}$$

The first term is given by

$$\begin{aligned}\mathrm{Var}\left(\sum_j (g_0(t_{ij}) - g_\lambda(t_{ij}))v_k(t_{ij})\right) &= m_i\,\mathrm{Var}(g_0(t_{11}) - g_\lambda(t_{11})v_k(t_{11}))\\ &\le m_i\,\mathrm{E}(g_0(t_{11}) - g_\lambda(t_{11})v_k(t_{11}))^2\\ &= m_i\int (g_0(t) - g_\lambda(t))^2 v_k(t)^2\,dt\end{aligned}$$

$$\leq m_i \|g_0 - g_\lambda\|^2 \sup_t v_k(t)^2$$
$$\leq c_0 m_i \tau_k^{-1} \|g_0 - g_\lambda\|^2 \leq c_0 m_i \tau_k^{-1} \lambda \|g_0\|_{\mathbb{K}}^2.$$

Note the last line follows from Lemma 4.5.1 and equation (4.5.7).

Turning to the second term, we have

$$\operatorname{Var}\left(\sum_j (Y_{ij} - g_\lambda(t_{ij}))v_k(t_{ij}) \middle| \mathcal{F}\right) = \sum_{j\ell} \operatorname{Cov}(Y_{ij}, Y_{i\ell}|\mathcal{F}) v_k(t_{ij}) v_k(t_{i\ell})$$
$$= \sum_{j\ell} (C(t_{ij}, t_{i\ell}) + \sigma^2 1_{j=\ell}) v_k(t_{ij}) v_k(t_{i\ell}).$$

When $j = \ell$ we use the assumed bounded variance and the orthonormality of the v_k to obtain

$$\mathrm{E}[(C(t_{ij}, t_{ij}) + \sigma^2) v_k(t_{ij})^2] = \int C(t,t) v_k(t)^2 \, dt + \sigma^2 \leq c_0.$$

When $j \neq \ell$ we use the definition of the covariance to obtain

$$\mathrm{E}[(C(t_{ij}, t_{i\ell}) v_k(t_{ij}) v_k(t_{i\ell})] = \int v_k(t) C(t,s) v_k(s)$$
$$= \langle v_k, C v_k \rangle = \mathrm{E}\langle X - g_0, v_k \rangle^2 \leq \mathrm{E}\langle X, v_k \rangle^2.$$

Using generic $\{c_i\}$ for the constants and recalling that m is the harmonic mean of the m_i we get the bound

$$\mathrm{E}\,\|\tilde{g}_\lambda - g_\lambda\|_\alpha^2 \leq \sum_{k=1}^\infty \frac{\tau_k^{2-\alpha}}{(\tau_k + \lambda)^2} \frac{1}{n^2} \sum_{i=1}^n \frac{1}{m_i^2} \left[\frac{c_0 m_i \lambda}{\tau_k} + m_i c_1 + m_i^2 \,\mathrm{E}\langle X, v_k \rangle^2\right] \quad (4.5.8)$$
$$= \sum_{k=1}^\infty \frac{\tau_k^{2-\alpha}}{(\tau_k + \lambda)^2} \frac{1}{n} \left[\frac{\lambda}{m\tau_k} c_0 + \frac{1}{m} c_1 + \mathrm{E}\langle X, v_k \rangle^2\right]. \quad (4.5.9)$$

We bound each term in the summand separately. If $\tau_k \asymp k^{-2r}$ and $\gamma > 1/2r$ is an

arbitrary number then we have

$$\sum_{k=1}^{\infty} \frac{\tau_k^{\gamma}}{(\tau_k+\lambda)^2} \asymp \int_0^{\infty} \frac{x^{-2r\gamma}}{(\lambda+x^{-2r})^2}\, dx = \int \frac{x^{2r(2-\gamma)}}{(\lambda x^{2r}+1)^2}\, dx.$$

Let $y = \lambda x^{2r}$ then $x = \lambda^{-1/2r} y^{1/2r}$ and $dx = \lambda^{-1/2r}(1/2r) y^{1/2r-1} dy$. Then the above becomes

$$\int \frac{\lambda^{-(2-\gamma)} y^{2-\gamma}}{(y+1)^2} \lambda^{-1/2r}(1/2r) y^{1/2r-1} dy = \frac{\lambda^{-(2-\gamma+1/2r)}}{2r} \int \frac{y^{1-\gamma+1/2r}}{(y+1)^2} dy$$

Notice the integral is finite as long as $\gamma > 1/2r$. We therefore have that, for any $\gamma > 1/2r$,

$$\sum_{k=1}^{\infty} \frac{\tau_k^{\gamma}}{(\tau_k+\lambda)^2} \asymp \lambda^{-(2-\gamma+1/2r)}. \tag{4.5.10}$$

Taking $\gamma = 1-\alpha$ and applying (4.5.10), which is greater than $1/2r$ as long as $\alpha < 1 - 1/2r$, the first term in (4.5.9) is given by

$$\sum_{k=1}^{\infty} \frac{\tau_k^{1-\alpha}}{(\tau_k+\lambda)^2} \frac{\lambda c_0}{nm} = O(\lambda^{-\alpha-1/2r}(nm)^{-1}).$$

Turning to the second term in (4.5.9), take $\gamma = 2-\alpha$ we have by the same arguments

$$\frac{c_2}{nm} \sum_k \frac{\tau_k^{2-\alpha}}{(\tau_k+\lambda)^2} \asymp (nm)^{-1} \lambda^{-\alpha-1/2r}.$$

Turning to the last term in (4.5.9) we can use the assumption that $\mathrm{E}\,\|X\|^2 < \infty$ to obtain

$$\sum_{k=1}^{\infty} \frac{\tau_k^{2-\alpha}}{(\tau_k+\lambda)^2} \frac{1}{n} \mathrm{E}\langle X, v_k\rangle^2 \le \mathrm{E}\,\|X\|^2 n^{-1} \max_k \frac{\tau_k^{2-\alpha}}{(\tau_k+\lambda)^2}.$$

Applying (4.5.6) with $\gamma = 2 - \alpha$ the above becomes

$$\mathrm{E}\,\|X\|^2 n^{-1} c_0 \lambda^{-\alpha}.$$

We thus conclude that

$$\|\tilde{g}_\lambda - g_\lambda\|_\alpha^2 = O_P\left((nm)^{-1}\lambda^{-\alpha-1/2r} + n^{-1}\lambda^{-\alpha}\right).$$

There will be two values of α that are especially important. The first is when $\alpha = 0$, which we use to bound the L^2 norm, while the second is for an arbitrary α that satisfies $1/2r < \alpha < 1 - 1/2r$, as this will be used to bound the last term in the next subSection.

Step 3: $\hat{g} - \tilde{g}_\lambda$

Recall that $\hat{g} = (T_{nm} + \lambda I)^{-1} h_{nm}$ and $\tilde{g}_\lambda = g_\lambda + (T + \lambda I)^{-1}(h_{nm} - T_{nm}(g_\lambda) - \lambda g_\lambda)$. Note that this also implies that $h_{nm} = (T_{nm} + \lambda I)\hat{g}$. So write

$$\begin{aligned}
\hat{g} - \tilde{g} &= \hat{g} - g_\lambda - (T + \lambda I)^{-1}(h_{nm} - T_{nm}(g_\lambda) - \lambda g_\lambda) \\
&= (T + \lambda I)^{-1}\left((T + \lambda I)(\hat{g} - g_\lambda) - (h_{nm} - (\lambda I + T_{nm})g_\lambda))\right) \\
&= (T + \lambda I)^{-1}\left((T + \lambda I)(\hat{g} - g_\lambda) - (T_{nm} + \lambda I)(\hat{g} - g_\lambda)\right).
\end{aligned}$$

Computing the α norm we can apply Parseval's and the definition of T_{nm} to obtain

$$\begin{aligned}
\|\hat{g} - \tilde{g}\|_\alpha^2 &= \sum_k \frac{\tau_k^{-\alpha}}{(\tau_k + \lambda)^2}\left[(\tau_k + \lambda)\langle \hat{g} - g_\lambda, v_k\rangle - \langle (T_{nm} + \lambda I)(\hat{g} - g_\lambda), v_k\rangle\right]^2 \\
&= \sum_k \frac{\tau_k^{2-\alpha}}{(\tau_k + \lambda)^2}\left[\langle \hat{g} - g_\lambda, v_k\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}(\hat{g}(t_{ij}) - g_\lambda(t_{ij}))v_k(t_{ij})\right]^2.
\end{aligned}$$

Notice that we can write $\hat{g}(t) - g_\lambda(t) = \sum_{\ell=1}^{\infty} h_\ell v_\ell(t)$ where $h_\ell = \langle \hat{g} - g_\lambda, v_\ell\rangle$.

We can then write

$$(\hat{g}(t_{ij}) - g_\lambda(t_{ij}))v_k(t_{ij}) = \sum_{\ell=1}^{\infty} h_\ell v_\ell(t_{ij})v_k(t_{ij}).$$

So the difference is given by

$$\begin{aligned}
&\langle \hat{g} - g_\lambda, v_k\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}(\hat{g}(t_{ij}) - g_\lambda(t_{ij}))v_k(t_{ij}) \\
&= h_k - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\sum_{\ell=1}^{\infty} h_\ell v_\ell(t_{ij})v_k(t_{ij}) \\
&= \sum_{\ell=1}^{\infty} h_\ell \left[\langle v_k, v_\ell\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} v_\ell(t_{ij})v_k(t_{ij})\right].
\end{aligned}$$

Let $\delta \in [0,1]$ be another constant similar, but potentially different from α. We can then apply CS to bound the above by

$$\begin{aligned}
|\langle \hat{g} - g_\lambda, v_k\rangle| &\le \left(\sum_{\ell=1}^{\infty}\frac{h_\ell^2}{\tau_\ell^\delta}\right)\sum_{\ell=1}^{\infty}\tau_\ell^\delta\left[\langle v_k, v_\ell\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} v_\ell(t_{ij})v_k(t_{ij})\right]^2 \\
&= \|\hat{g} - g_\lambda\|_\delta^2 \sum_{\ell=1}^{\infty}\tau_\ell^\delta\left[\langle v_k, v_\ell\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} v_\ell(t_{ij})v_k(t_{ij})\right]^2.
\end{aligned}$$

To get the asymptotic order of the summation term above, by Markov's inequality, it is enough to bound its expected value (since it is positive). Taking the expected value of the summation we get that

$$\begin{aligned}
&\sum_{\ell=1}^{\infty}\tau_\ell^\delta \,\mathrm{E}\left[\langle v_k, v_\ell\rangle - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} v_\ell(t_{ij})v_k(t_{ij})\right]^2 \\
&= \sum_{\ell=1}^{\infty}\frac{\tau_\ell^\delta}{nm}\,\mathrm{Var}(v_\ell(t_{11})v_k(t_{11})) \\
&\le \sum_{\ell=1}^{\infty}\frac{\tau_\ell^\delta}{nm}\int v_\ell(t)^2 v_k(t)^2\,dt \le \sum_{\ell=1}^{\infty}\frac{\tau_\ell^\delta}{nm}\sup_t v_k(t)^2\int v_\ell(t)^2\,dt \le \sum_{\ell=1}^{\infty}\frac{c_0\tau_\ell^\delta}{nm\tau_k}.
\end{aligned}$$

Recall that $\tau_\ell \asymp \ell^{-2r}$, so the above sum is finite as long as $\delta > 1/2r$. Putting everything together and applying (4.5.6) we have the bound

$$\|\hat{g} - \tilde{g}\|_\alpha^2 \leq O_P(1)\|\hat{g} - g_\lambda\|_\delta^2 \frac{c_0}{nm} \sum_k \frac{\tau_k^{1-\alpha}}{(\tau_k + \lambda)^2} \asymp O_P(1)\|\hat{g} - g_\lambda\|_\delta^2 (nm)^{-1}\lambda^{-\alpha-1/2r},$$

which holds for any $0 \leq \alpha < 1 - 1/2r$ and any $\delta > 1/2r$.

Assume that λ is such that $(nm)^{-1}\lambda^{-\alpha-1/2r} \to 0$, then it follows that $\|\hat{g} - \tilde{g}\|_\alpha^2 = o_P(\|\hat{g} - g_\lambda\|_\delta^2)$. A triangle inequality gives

$$\|\tilde{g}_\lambda - g_\lambda\|_\delta \geq \|\hat{g} - g_\lambda\|_\delta - \|\hat{g} - \tilde{g}\|_\delta = (1 + o_P(1))\|\hat{g} - g_\lambda\|_\delta.$$

This implies that

$$\|\hat{g} - g_\lambda\|_\delta = O_P(\|\tilde{g}_\lambda - g_\lambda\|_\delta).$$

Finally, take $\alpha = 0$ and $\delta > 1/2r$ then we have that

$$\begin{aligned}\|\hat{g} - \tilde{g}\|^2 &= O_P(1)(nm)^{-1}\lambda^{-1/2r}\|\tilde{g}_\lambda - g_\lambda\|_\delta^2 \\ &= O_P(1)(nm)^{-1}\lambda^{-1/2r}[(nm)^{-1}\lambda^{-\delta-1/2r} + n^{-1}\lambda^{-\delta}]\end{aligned}$$

If we assume that λ is such that $(nm)^{-1}\lambda^{-\delta-1/2r} \to 0$ then the above simplifies to

$$o_P(1)\lambda^\delta[(nm)^{-1}\lambda^{-\delta-1/2r} + n^{-1}\lambda^{-\delta}] = o_P(1)[(nm)^{-1}\lambda^{-1/2r} + n^{-1}],$$

as desired.

Note that in the last paragraph, we made a more explicit assumption about how quickly λ tends to zero. Note that the optimal rate is $\lambda = (nm)^{2r/(1+2r)}$. For this value of λ we have that $(nm)^{-1}\lambda^{-\alpha-1/2r} \to 0$ for any value of $\alpha < 1$ since $1 + 1/2r = (2r + 1)/2r$.

Bibliography

Ahonen, T., Hadid, A. and Pietikainen, M. Ahonen *et al.* (2006). Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, **28(12),** 2037–2041.

Aydin, B., G. Pataki, H. Wang, Bullitt, E. and Marron, J. S. Aydin *et al.* (2009). A principal component analysis for trees.

Belkin, M. and Niyogi, P. Belkin and Niyogi (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, **15,** 1373–1396.

Bohrer, R. Bohrer (1967). On sharpening scheffe bounds. *Journal of the Royal Statistical Society. Series* B, **29(1),** 110–114.

Bunea, F., Hoff, P., Holmes, C., Kim, P., Koltchinskii, V., Lafferty, J., Lerman, G., van de Geer, S., Wegkamp, M. and Yu, B. Bunea *et al.* (2014). Low-dimensional structure in high-dimensional systems. Technical Report. SAMSI.

Cao, X., Wei, Y., Wen, F. and Sun, J. Cao *et al.* (2014). Face alignment by explicit shape regression.

Chen, D. and Müller, H.G. Chen and Müller (2012). Nonlinear manifold representations for functional data. *The Annals of Statistics*, **40,** 1–29.

Chen, K. and Müller, H.G. Chen and Müller (2012). Conditional quantile analysis when covariates are functions, with application to growth data. *Journal of the Royal Statistical Society, Series* B, **74,** 67–89.

Chenouri, S. E., Kobelevskiy, P. and Small, C. G. Chenouri *et al.* (2015). Spanifold: spanning tree flattening onto lower dimension. *Stat*, **4,** 15–31.

Choi, H. and Reimherr, M. Choi and Reimherr (2018). A geometric approach to confidence regions and bands for functional parameters.

Claes, P., Hill, H. and Shriver, M. D. Claes *et al.* (2014a). Toward dna-based facial composites: preliminary results and validation. *Forensic Sci Int Genet*, **13,** 208–16.

Claes, P., Liberton, D. K., Daniels, K., Rosana, K. M., Quillen, E. E., Pearson, L. N., McEvoy, B., Bauchet, M., Zaidi, A. A., Yao, W., Tang, H., Barsh, G. S., Absher, D. M., Puts, D. A., Rocha, J., Belez, S., Pereira, R. W., Baynam, G., Suetens, P., Vandermeulen, D., Wagner, J. K., Boster, J. S. and Shriver, M. D. Claes *et al.* (2014b). Modeling 3d facial shape from dna. *PLoS Genet*, **10(3)**.

Claes, P., Walters, W. and Clement, J. Claes *et al.* (2012). Improved facial outcome assessment using a 3d anthropometric mask. *International Journal of Oral and Maxillofacial Surgery*, **41(3),** 324–330.

Dimeglio, C., Gallón, S., Loubes, J. and Maza, E. Dimeglio *et al.* (2014). A robust algorithm for template curve estimation based on manifold embedding. *Computational Statistics and Data Analysis*, **70,** 373–386.

Drira, H., Amor, B. Ben, Daoudi, M., Srivastava, A. and Berretti, S. Drira *et al.* (2012). 3d dynamic expression recognition based on a novel deformation vector field and random forest. *ICPR*, 1104–1107.

Duchesne, P. and Lafaye de Micheaux, P. Duchesne and Lafaye de Micheaux (2010). Computing the distribution of quadratic forms: Further comparisons between the liu-tang-zhang approximation and exact methods. *Computational Statistics and Data Analysis*, **54,** 858–862.

Elhamifar, E. and Vidal, R. Elhamifar and Vidal (2011). Sparse manifold clustering and embedding. In *Advances in neural information processing systems*, pp. 55–63. NIPS Foundation.

Ellingson, L., Patrangenaru, V. and Ruymgaart, F. Ellingson *et al.* (2013). Nonparametric estimation of means on Hilbert manifolds and extrinsic analysis of mean shapes of contours. *Journal of Multivariate Analysis*, **122,** 317–333.

Ettinger, B., Perotto, S. and Sangalli, L. M. Ettinger *et al.* (2016). Spatial regression models over two-dimensional manifolds. *Biometrika*, **103(1),** 71–88.

Fletcher, P. T. Fletcher (2013). Geodesic regression and the theory of least squares on riemannian manifolds.

Golub, G. H., Heath, M. and Wahba, G. Golub *et al.* (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21(2),** 215–223.

Graves, S., Hooker, G. and Ramsay, J. Graves *et al.* (2009). *Functional data analysis with r and matlab*. Springer.

Gromenko, O. and Kokoszka, P. Gromenko and Kokoszka (2013). Nonparametric inference in small data sets of spatially indexed curves with application to ionospheric trend determination. *Computational Statistics and Data Analysis*, **59,** 82–94.

Ha, H.T. and Provost, S.B. Ha and Provost (2011). An accurate approximation to the distribution of a linear combination of non-central chi-square random variables. *REVSTATStatistical Journal*, **11(3),** 231–254.

Hammond, P., Hutton, T. J., Allanson, J. E., Buxton, B., Campbell, L. E., Clayton-Smith, J., Donnai, D., Karmiloff-Smith, A., Metcalfe, K., Murphy, K. C., Patton, M., Barbara, P., Prescott, K., Scambler, P., Shaw, A., Smith, A. C. M., Stevens, A. F., Temple, I. K., Hennekam, R. and Tassabehji, M. Hammond *et al.* (2005). Discriminating power of localized three-dimensional facial morphology. *The American Journal of Human Genetics*, **77(6),** 999–1010.

Horváth, L. and Kokoszka, P. Horváth and Kokoszka (2012). *Inference for Functional Data with Applications*. Springer, New York.

Huang, D., Ding, H., Wang, C., Wang, Y., Zhang, G. and Chen, L. Huang *et al.* (2014). Local circular patterns for multi-modal facial gender and ethnicity classification. *Image and Vision Computing*, **32:12,** 1181–1193.

Huang, D., Sun, J., Yang, X., Weng, D. and Wang, Y. Huang *et al.* (2014). 3d face analysis: Advances and perspectives. *Chinese Conference on Biometric Recognition*, 1–21.

Imhof, J. P. Imhof (1961). Computing the distribution of quadratic forms in normal variables. *Biometrika*, **48,** 419–426.

Jain, A.K. and Li, S.Z. Jain and Li (2011). *Handbook of face recognition*. New York: springer.

Karhunen, K. Karhunen (1946). Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae. Series AI*, **34**.

Kato, T. Kato (1966). *Perturbation theory for linear operators*. Springer Science & Business Media.

Kokoszka, P. and Reimherr, M. Kokoszka and Reimherr (2012). Determining the order of the functional autoregressive model. *Journal of Time Series Analysis*, **34,** 116–129.

Kokoszka, P. and Reimherr, M. Kokoszka and Reimherr (2017). *Introduction to functional data analysis*. Chapman and Hall/CRC.

Kurtek, S. and Drira, H. Kurtek and Drira (2015). A comprehensive statistical framework for elastic shape analysis of 3d faces. *Computers and Graphics*, **51,** 52–59.

Lee, J.M. Lee (2003). *Introduction to smooth manifold*. Springer New York.

Lila, E. and Aston, J. A. D. Lila and Aston (2017). Functional and geometric statistical analysis of textured surfaces with an application to medical imaging.

Lila, E., Aston, J. A. D. and Sangalli, L. M. Lila *et al.* (2016). Smooth principal component analysis over two-dimensional manifolds with an application to neuroimaging. *The Annals of Applied Statistics*, **10(4),** 1854–1879.

Lindgren, F. and Rue, H. Lindgren and Rue (2013). Bayesian spatial and spatiotemporal modeling with r-inla. *Journal of Statistical Software*, **63**.

Loéve, M. Loéve (1946). Functions aleatoire de second ordre. *Revue science*, **84,** 195–206.

Marron, J. S. Marron (2014). Object oriented data analysis: Open problems regarding manifolds.

Marron, J. S. and Alonso, A. M. Marron and Alonso (2014). Overview of object oriented data analysis.

Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. Nadler *et al.* (2006). Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, **21,** 113–127.

Patrangenaru, V. and Ellingson, L. Patrangenaru and Ellingson (2015). *Nonparametric statistics on manifolds and their applications to object data analysis*. CRC Press.

Pazouki, M. and Schaback, R. Pazouki and Schaback (2011). Bases for kernel-based spaces. *Journal of Computational and Applied Mathematics*, **236(4),** 575–588.

Pigoli, D., Z.Hadjipantelis, P., Coleman, J. S. and Aston, J. A. Pigoli *et al.* (2018). The analysis of acoustic phonetic data: exploring differences in the spoken romance languages. *Journal of the Royal Statistical Society. Series* C, **67(4),** 1–27.

Porro-Muñoz, D., Silva-Mata, F. J., Revilla-Eng, A., Talavera-Bustamante, I. and Berretti, S. Porro-Muñoz *et al.* (2014). 3d face recognition by functional data analysis. *Iberoamerican Congress on Pattern Recognition*, 818–826.

Ramsay, J. O. Ramsay (2006). *Functional data analysis*. John Wiley & Sons, Inc.

Ramsay, J. O. and Dalzell, C. J. Ramsay and Dalzell (1991). Some tools for functional data analysis. 539–572.

Ramsay, J. O. and Silverman, B. W. Ramsay and Silverman (2007). *Applied functional data analysis: methods and case studies*. Springer.

Ramsay, J.O. and Silverman, B.W. Ramsay and Silverman (2005). *Functional data analysis*. Springer.

Ramsay, T. Ramsay (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64,** 307–319.

Reiss, P.T., Mennes, M., Petkova, E., Huang, L., Hoptman, M.J., Biswal, B.B., Colcombe, S.J., Zuo, X.-N. and Milham, M.P. Reiss *et al.* (2011). Extracting information from functional connectivity maps via function-on-scalar regression. *NeuroImage*, **56,** 140–148.

Rohlf, F. J. and Slice, D. Rohlf and Slice (1990). Extensions of the procrustes method for the optimal superimposition of landmarks. *Systematic Biology*, **39(1)**.

Saul, L. K. and Roweis, S. T. Saul and Roweis (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, **4,** 119–155.

Shen, D., Shen, H., Bhamidi, S., Maldonado, Y. M., Kim, Y. and Marron, J. S. Shen *et al.* (2014a). Functional data analysis of tree data objects.

Skwerer, S., Bullitt, E., Huckemann, S., Miller, E., Oguz, I., Owen, M., Patrangenaru, V., Provan, S. and Marron, J. S. Skwerer *et al.* (2014b). Tree-oriented analysis of brain artery structure.

Taigman, Y., Yang, M., Ranzato, M.A. and Wolf, L. Taigman *et al.* (2014). Deepface: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1708.

Tang, R. and Müller, H.G. Tang and Müller (2009). Time-synchronized clustering of gene expression trajectories. *Biostatistics*, **10,** 32–45.

Tenenbaum, J.B., de Silva, V. and Langford, J. C. Tenenbaum *et al.* (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, **290,** 2319–2323.

Turk, M. A. and Pentland, A. P. Turk and Pentland (1991). Face recognition using eigenfaces. *Computer Vision and Pattern Recognition*.

Ullah, S. and Finch, C. F. Ullah and Finch (2013). Applications of functional data analysis: A systematic review. 13–43.

Verzelen, N., Tao, W. and Müller, H.G. Verzelen *et al.* (2012). Inferring stochastic dynamics from functional data. *Biometrika*, **99,** 533–550.

Wang, H. and Marron, J. S. Wang and Marron (2007). Object oriented data analysis: Sets of trees.

Wang, J. L., Chiou, J. M. and Mueller, H. G. Wang *et al.* (2016). Functional data analysis. *Annual Review of Statistics and Its Application*, 257–295.

Wang, Y., Marron, J. S., Aydin, B., Ladha, A., Bullitt, E. and Wang, H. Wang *et al.* (2012). A nonparametric regression model with tree-structured response.

Xia, B., Amor, B. B., Daoudi, M. and Drira, H. Xia *et al.* (2014). Can 3d shape of the face reveal your age? *Computer Vision Theory and Applications (VISAPP), 2014 International Conference*, **2,** 5–13.

Xia, B., Amor, B. B., Daoudi, M. and L.Ballihi Xia *et al.* (2013). Gender and 3d facial symmetry: What's the relationship? *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*.

Yang, Y. and Dunson, D. B. Yang and Dunson (2016). Bayesian manifold regression.

Yuan, M. and Cai, T. T. Yuan and Cai (2010). A reproducing kernel hilbert space approach to functional linear regression. *The Annals of Statistics*, **38(6),** 3412–3444.

Zhang, X. and Wang, J.L. Zhang and Wang (2016). From sparse to dense functional data and beyond. *The Annals of Statistics*, **44(5),** 2281–2321.

Zhang, Z. and Zha, H. Zhang and Zha (2004). Principal manifolds and nonlinear dimension reduction via tangent space alignment. *SIAM Journal of Scientific Computing*, **26,** 313–338.

Zipunnikov, V., Caffo, B., Yousem, D.M., Davatzikos, C., Schwartz, B.S. and Crainiceanu, C. Zipunnikov *et al.* (2011). Functional principal component model for high-dimensional brain imaging. *NeuroImage*, **58,** 772–784.

Vita

Hyun Bin Kang

Research Interests

Functional Data Analysis; High-Dimensional Statistical Inference; Manifold Learning; Shape Analysis; Spatial Statistics; Variable Selection and Feature Screening

Education

Ph.D. in Statistics, The Pennsylvania State University, 2018.
(Dissertation Advisor: Matthew Reimherr)
B.S. in Statistics and Mathematical Economic Analysis, Rice University, 2012.

Publications

Kang, H., Reimherr, M., Shriver, M. and Claes, P. (2018). A Functional Approach to Manifold Data Analysis with an Application to High-Resolution 3D Faces. *Submitted*.
Zimmerman, W. A., **Kang, H.**, Kim, K., Gao, M., Johnson, G., Clariana, R. and Zhang. (2018). Computer-Automated Approach for Scoring Short Essays in an Introductory Statistics Course. *Revised and resubmitted to Journal of Statistics Education*.

Honors and Awards

Award for Excellence in Online Instruction, 2016.
Jack and Eleanor Pettit Scholarship in Science, 2016.
Academic Computing Fellowship, 2016–2018.
Award for Distinguished Masters Qualifying Exam, 2014.
Brumbach Distinguished Graduate Fellowship, 2013–2014.