FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

FUNCTIONAL COMPONENT ANALYSIS AND REGRESSION USING ELASTIC METHODS

By

J. DEREK TUCKER

A Dissertation submitted to the
Department of Statistics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Degree Awarded:
Summer Semester, 2014

J. Derek Tucker defended this dissertation on May 20, 2014.
The members of the supervisory committee were:

Anuj Srivastava

Professor Co-Directing Dissertation

Wei Wu

Professor Co-Directing Dissertation

Eric Klassen

University Representative

Fred Huffer

Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

I dedicate this work to my wife and my children. Your unconditional support, love, and patience made this work possible and allowed me to achieve my dreams.

ACKNOWLEDGMENTS

I would like to acknowledge the support of my colleagues at the Naval Surface Warfare Center Panama City Division (NSWC PCD) in the pursuit of my PhD degree. It has been a great honor to further my education while employed at the lab. I am thankful for the support of Dr. Frank Crosby and Dr. Quyen Huynh, at NSWC PCD who supported me and introduced me to my advisor, Dr. Anuj Srivastava. Furthermore, I would like to thank my advisors, Dr. Anuj Srivastava and Dr. Wei Wu for their continual support, patience, technical discussions, and dedication to my success. From them I have learned and accomplished much that will benefit me in my future endeavors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Constructing generative models for functional observations is an important task in statistical function analysis. In general, functional data contains both phase (or x or horizontal) and amplitude (or y or vertical) variability. Traditional methods often ignore the phase variability and focus solely on the amplitude variation, using cross-sectional techniques such as functional principal component analysis for dimensional reduction and regression for data modeling. Ignoring phase variability leads to a loss of structure in the data, and inefficiency in data models. Moreover, most methods use a "pre-processing" alignment step to remove the phase-variability; without considering a more natural joint solution.

This dissertation presents three approaches to this problem. The first relies on separating the phase (x-axis) and amplitude (y-axis), then modeling these components using joint distributions. This separation in turn, is performed using a technique called *elastic alignment of functions* that involves a new mathematical representation of functional data. Then, using individual principal components, one for each phase and amplitude components, it imposes joint probability models on principal coefficients of these components while respecting the nonlinear geometry of the phase representation space. The second combines the phase-variability into the objective function for two component analysis methods, functional principal component analysis and functional principal least squares. This creates a more complete solution, as the phase-variability is removed while simultaneously extracting the components. The third approach combines the phase-variability into the functional linear regression model and then extends the model to logistic and multinomial logistic regression. Through incorporating the phase-variability a more parsimonious regression model is obtained and therefore, more accurate prediction of observations is achieved. These models then are easily extended from functional data to curves (which are essentially functions in $\mathbb{R}^2$) to perform regression with curves as predictors.

These ideas are demonstrated using random sampling for models estimated from simulated and real datasets, and show their superiority over models that ignore phase-amplitude separation. Furthermore, the models are applied to classification of functional data and achieve high performance in applications involving SONAR signals of underwater objects, handwritten signatures, periodic body movements recorded by smart phones, and physiological data.

CHAPTER 1

INTRODUCTION

The statistical analysis of functional data is fast gaining prominence in the statistics community as this kind of "big data" is central to many applications. For instance, functional data can be found in a broad swath of application areas ranging from biology, medicine, chemistry, geology, sports, and financial analysis. One can easily encounter a problem where the observations are real-valued functions on an interval, and the goal is to perform their statistical analysis.

By statistical analysis we mean *to compare, align, average, and model* a collection of random observations. Figure 1.1 presents examples of functional data with the Panels a - c corresponding to the average temperature in Celsius measured at 35 distinct sites in Canada, the growth velocity of girls, and the x-gyorometer measurement of an iPhone 4 while riding a bicycle. The analysis of such data has been of great interest and solutions have been proposed using the standard $\mathbb{L}^2$ metric to compute distances, cross-sectional (i.e., point-wise) means and variances, and principal components of the observed functions [52].

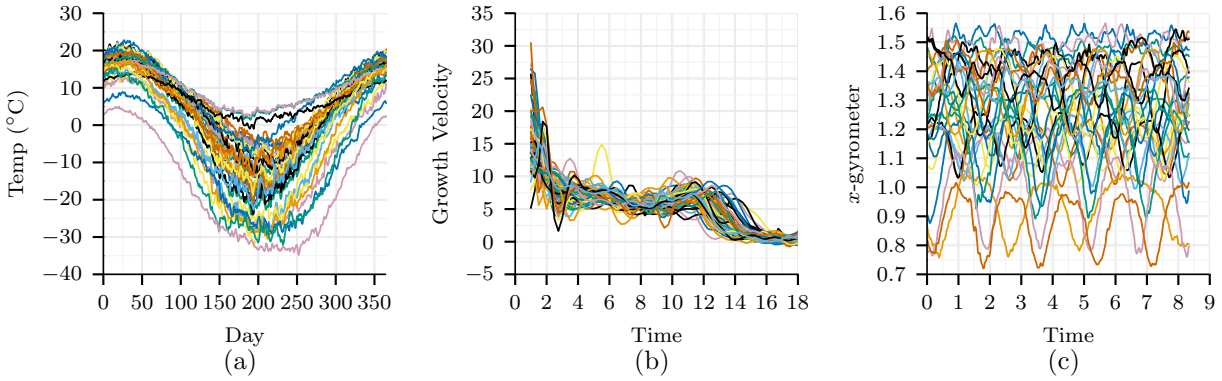


Figure 1.1: Examples of functional data which includes (a) the average Canadian temperature measured at 35 different sites over 365 days, (b) the growth velocity for 21 different girls, and (c) the gyrometer in the x direction measured using a iPhone 4 while riding a bicycle for 30 subjects.

Questions then arise on how can we model the functions. Can we use the functions to classify diseases? Or can we use them as predictors in a regression model? One problem occurs when performing these type of analyses is that functional data can contain variability in time (x-direction) and amplitude (y-direction) which complicates the analysis. The problem then becomes how do we account for and handle this variability in the models that are constructed from functional data.

1.1 Motivation

Since the approach in functional data analysis treats the entire function as the unit of observation (as opposed to multivariate analysis which works with a vector of measurements), dimension reduction becomes an important issue. Strictly speaking, since functional data intrinsically are infinite dimensional, the process of dimension reduction becomes essential. There has been a great deal of research devoted to dimension reduction in high-dimensional multivariate data (e.g., principal component analysis (PCA) [22], partial least squares (PLS) [73], and multidimensional scaling [66]).

Dimension reduction in functional data has focused mainly on PCA [52], canonical correlation analysis [42], and regression type problems [9, 50]. Specifically, these methods have focused on extending the standard multivariate methods to functional data using the $\mathbb{L}^2$ framework. Moreover, they implicitly assume that the observed functions are already temporally aligned and all the variability is restricted only to the y-axis (or the vertical axis). By temporal alignment we mean corresponding peaks and valleys line up along the x-axis (or the horizontal axis).

A serious challenge arises when functions are observed with flexibility or domain warping along the x-axis, which in reality is quite common. This warping may come either from an uncertainty in the measurement process, or may simply denote an inherent variability in the underlying process itself. The warping then needs to be separated from the variability along the y-axis. An interesting aspect of functional data is that the underlying variability can be ascribed to two sources. The variability exhibited in functions after alignment is termed the amplitude (or y or vertical) variability and the warping functions that are used in the alignment are said to capture the phase (or x or horizontal) variability. A more explicit mathematical definition of amplitude- and phase-variability will be made in Chapter 2.

When the phase-variability is ignored the analysis can be misleading and incorrect models can be constructed. A prominent example of this situation is functional principal component

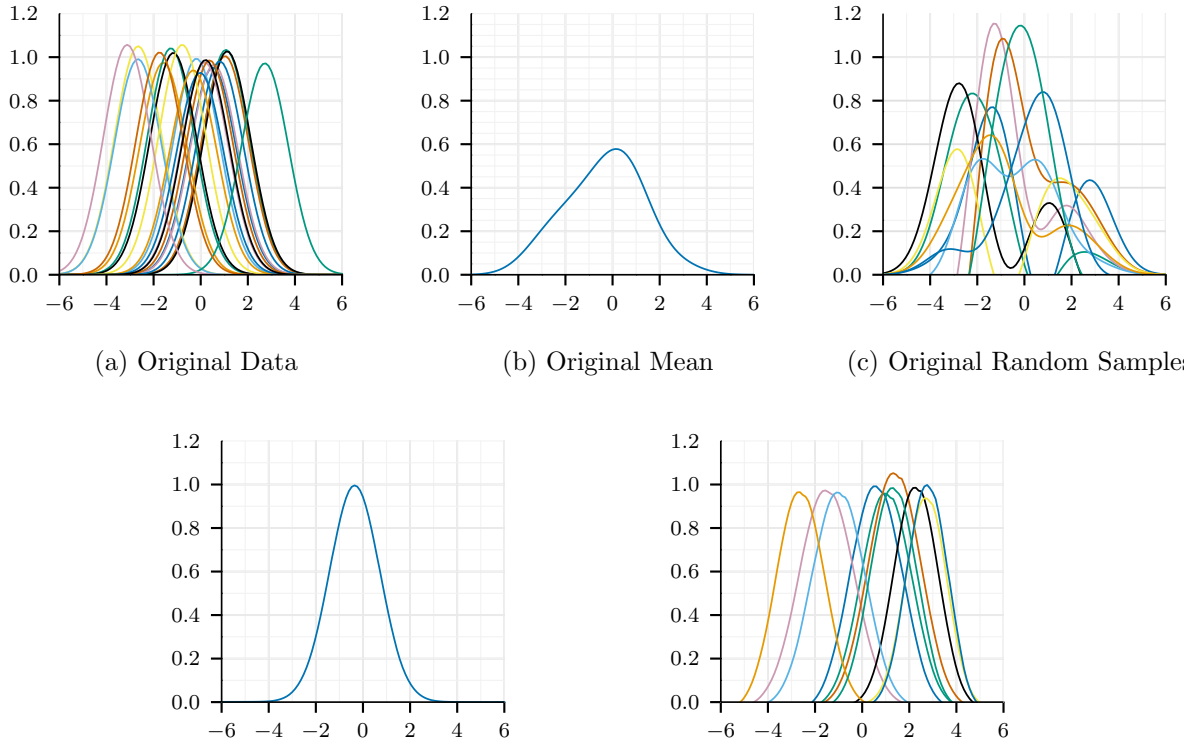
analysis (fPCA) [52] which is used to discover dominant modes of variation in the data and has been extensively used in modeling functional observations. If the phase-variability is ignored, the resulting model may fail to capture patterns present in the data and will lead to inefficient data models.

Fig. 1.2 provides an illustration of this using simulated functional data. This data was simulated using the equation $y_i(t) = z_i e^{-(t-a_i)^2/2}$, $t \in [-6, 6]$, $i = 1, 2, \ldots, 21$, where z_i is *i.i.d.* $\mathcal{N}(1, (0.05)^2)$ and a_i is *i.i.d.* $\mathcal{N}(0, (1.25)^2)$. The top-left plot shows the original data, each sample function here is a unimodal function with slight variability in height and a large variability in the peak placement. One can attribute different locations of the peak to the phase-variability. If one takes the cross-sectional mean of this data, ignoring the phase-variability, the result is shown in the top-middle plot. The unimodal structure is lost in this mean function with large amounts of stretching. Furthermore, if one performs fPCA on this data and imposes the standard independent normal models on fPCA coefficients (details of this construction are given later), the resulting model will lack this unimodal structure. Shown in the top-right plot are random samples generated from such a probability model on the function space where a Gaussian model is imposed on the fPCA coefficients. These random samples are not representative of the original data; the essential shape of the function is lost, with some of the curves having two, three, or even more peaks.

The reason why the underlying unimodal pattern is not retained in the model is that the phase variability was ignored. We argue that a proper technique is to incorporate the phase and amplitude variability into the model construction, which in turn incorporates into the component or regression analysis. While postponing details for later, we show results obtained by a separation-based approach in the bottom row. The mean of the aligned functions is shown in the bottom left panel of Fig. 1.2. Clearly retained is the unimodal structure, and the random samples generated under a framework that model the phase and amplitude variables individually have the same structure. Some random samples are shown in the bottom right panel; these displays are simply meant to motivate the framework, the mathematical details are provided later in the dissertation. This example clearly motivates the need for the inclusion of warping for modeling functional data that contains phase-variability.

Research in functional data analysis has a wide breadth across multiple areas that include:

- Fitting function to discrete data



(a) Original Data (b) Original Mean (c) Original Random Samples

(d) Aligned Mean (e) Samples from Proposed Models

Figure 1.2: Samples drawn from a Gaussian model fitted to the principal components for the un-aligned and aligned data.

- Summary statistics
- Dimension reduction
- Clustering and classification
- Registration
- Component analysis
- Regression
- Developing functional priors for Bayesian analysis
- Modeling

In most of these areas, the functions are assumed to be aligned and the phase-variability problem is ignored or solved using a registration component. In each of these areas, one can find data that

is warped and a more complete solution is needed. In this dissertation, we focus on the areas of clustering and classification, component analysis, and regression with respect to warped data.

1.2 Contributions

We propose three approaches to the phase-variability problem. First, one can simply align the functional data prior to performing a component analysis tool. After alignment we have the separated phase and amplitude components where we can estimate the sample means and covariance on the phase and amplitude components, respectively. From the estimated summary statistics, we can perform a component analysis method (e.g., fPCA) on the phase and amplitude. Then we model the original data using joint Gaussian or non-parametric models on the dimension-reduced representations. Second, the phase-variability can be incorporated into the component analysis to create a warping-invariant or "elastic" component analysis. In this approach the objective function for the component analysis method is modified to include a phase-variability term. This in turn alters how the optimization is performed and the data is aligned while the corresponding components are extracted. Third, the phase-variability can be incorporated into the regression model to create a warping-invariant or "elastic" regression model. Similar to the second approach, the regression model is modified to include a phase-variability term. This in turn alters how the the model is identified and the data is aligned while the corresponding model coefficients are computed. The last two approaches provide a more natural approach to the analysis as the alignment is not a "pre-processing" step, but rather part of the complete solution.

The remainder of this dissertation is organized as follows. Chapter 2 reviews functional data analysis methods and various alignment methods for function data. Specifically, we review the elastic method based on the extended Fisher-Rao metric of [40, 63]. The Fisher-Rao method provides a novel transformation of the data, which then provides a proper distance, thus satisfying all desired properties in alignment, such as symmetry (optimal alignment of f to g is same as that of g to f), positive definiteness (the cost term between any two functions is nonnegative and it equals zero if and only if one can be perfectly aligned to the other), and the triangle inequality. Given this theoretical superiority over other published methods, we will use this method in the construction of our analysis. Chapter 3 presents performing modeling and classification using fPCA on the phase and amplitude components extracted from the alignment of the data and presents results using

several data sets which include a simulated data set, a signature data set from [75] , an iPhone action data set from [44], and a SONAR data set. Chapter 4 then develops the method of joint alignment and component analysis. We will focus on two methods: fPCA and functional partial least squares (fPLS). In each of these settings we will construct an objective function that includes phase-variability, and construct methods for optimizing the objective functions with results demonstrated on the iPhone action data set and the gait data from [52]. Chapter 5 develops the joint alignment and functional linear regression model. We will then use this model to develop the functional logistic and multinomial logistic regression models; and provide results on the classification of biomedical data. In Chapter 6, we will extend the regression models from Chapter 5 to use open curves in $\mathbb{R}^2$ as predictors and demonstrate results on the classification of shapes. Lastly, Chapter 7 provides concluding remarks and future research directions.

CHAPTER 2

LITERATURE REVIEW

In this chapter we summarize the state of the art in functional data analysis and previous work on phase and amplitude separation of functional data. We will provide a review of the current work in functional-based component analysis and functional regression. Finally, we will provide a brief description of the elastic method for functional alignment and motivation on our use of this framework throughout this work.

2.1 Functional Data Analysis

The goal of functional data analysis is to represent, display, study, explain, and compare functions. In other words, it is the same goal of any area of statistical study; as we would like to discover ways to represent the data in such a way that we can study the patterns of variation, compute statistics, and generate models of the data. The work of [23, 47, 50, 52, 54] have extended tools from multivariate statistics or developed new tools to accomplish this goal. We will describe the summary statistics of functional data, smoothing of functional data, principal component analysis, and linear regression models using this type of data. For a more complete review of this topic the reader is referred to [52].

2.1.1 Summary Statistics

Let f be a real-valued function with the domain $[0, 1]$; the domain can easily be transformed to any other interval. For concreteness, only functions that are absolutely continuous on $[0, 1]$ will be considered; let $\mathcal{F}$ denote the set of all such functions. In practice, since the observed data is discrete, this assumption is not a restriction. Assume that we have a collection of functions, $f_i(t)$, $i = 1, \ldots, N$ and we wish to calculate statistics on this set. The classical summary statistics apply to function data in a cross-sectional manner. The mean function is constructed using

$$\bar{f}(t) = \frac{1}{N} \sum_{i=1}^{N} f_i(t)$$

which is the average of the set of functions, $\{f_i\}$ point-wise across the replications. We can define the variance function similarly as

$$\text{var}(f(t)) = \frac{1}{N-1}\sum_{i=1}^{N}\left(f_i(t) - \bar{f}(t)\right)^2$$

and the standard deviation function is the point-wise square root of the variance function.

The covariance function summarizes the dependence of functions across different time values and is computed for all t_1 and t_2

$$\text{cov}_f(t_1, t_2) = \frac{1}{N-1}\sum_{i=1}^{N}\left(f_i(t_1) - \bar{f}(t_1)\right)\left(f_i(t_2) - \bar{f}(t_2)\right).$$

The correlation function is then computed using

$$\text{corr}_f(t_1, t_2) = \frac{\text{cov}_f(t_1, t_2)}{\sqrt{\text{var}(f(t_1))\,\text{var}(f(t_2))}}.$$

All of these summary statistical functions are the functional analogues of the covariance and correlation matrices in multivariate statistical analysis. On that note, if we have two sets of functions f_i and g_i, $i = 1, \ldots, N$ we can calculate a cross-covariance function

$$\text{cov}_{f,g}(t_1, t_2) = \frac{1}{N-1}\sum_{i=1}^{N}\left(f_i(t_1) - \bar{f}(t_1)\right)\left(g_i(t_2) - \bar{g}(t_2)\right),$$

and the cross-correlation function

$$\text{corr}_{f,g}(t_1, t_2) = \frac{\text{cov}_{f,g}(t_1, t_2)}{\sqrt{\text{var}(f(t_1))\,\text{var}(g(t_2))}}.$$

2.1.2 Smoothing Functional Data

As assumed earlier the construction of functional observations, f_i is done using discrete data and this process is done separately or independently for each observation. In some situations, the signal-to-noise ration is low, or the data is noisy, or the discrete observations are few in number. This can cause problems in generating sufficient summary statistics and it can be essential to use information in neighboring or similar functions to get better estimates. Therefore, we seek methods to smooth the functions and interpolate between samples to improve the statistical estimates.

Representing functions by basis functions. We can represent a function by a set of basis functions, θ_i, $i = 1, \ldots, k$. An example of types of basis can be the power series, Fourier series, and B-spline basis functions. We can represent a function f by a linear expansion

$$f(t) = \sum_{i=1}^{p} b_i \theta_i$$

in terms of p basis functions θ_i and coefficients b_i. By setting the dimension of the basis expansion p we can control the degree to which the functions are smoothed. If we choose p large enough we can fit the data well, but might fit any additive noise. If we choose p too small we remove structure from the data that might be important to the analysis. Therefore, the choice of p is important in determining the amount of smoothing desired. It should be noted that there is not one universally acceptable basis for functional data analysis. The type of data and problem application will dictate which basis is best for the problem. For example, one might choose the Fourier basis if the data is periodic as the basis is periodic and will generally fit the data well.

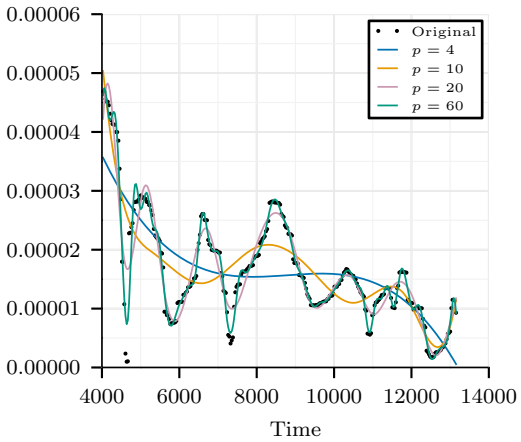


Figure 2.1: Example of smoothing of functional data by changing the number of basis elements.

We can find the basis coefficients, b_i by minimizing the least-squares criterion

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} (\mathbf{f} - \Theta\mathbf{b})^\mathsf{T} (\mathbf{f} - \Theta\mathbf{b})$$

where $\mathbf{f}$ is a vector containing the values of the function f_i, Θ is a matrix whose columns are the basis functions θ_i, and $\mathbf{b} = [b_1, \ldots, b_p]$. The solution to this problem is the standard least squares solution

$$\mathbf{b}^* = (\Theta^\mathsf{T}\Theta)^{-1}\Theta^\mathsf{T}\mathbf{f}.$$

We can also use a weighted least squares approach using a matrix W as the weights,

$$\mathbf{b}^* = (\Theta^\mathsf{T} W \Theta)^{-1} \Theta^\mathsf{T} W \mathbf{f}.$$

In some applications this might be desirable if the noise added to the function is known and the covariance matrix, Σ_n of the noise can be calculated and we therefore can set $W = \Sigma_n$. Fig. 2.1 presents an example smoothing a measured SONAR signal [69] while increasing the number of basis elements p. A B-spline basis was used and as p increases we observe a better fit of the data and eventually over-fitting to the noise, as p decreases more smoothing occurs.

Kernel Smoothing. Another type of smoothing of functional data is kernel smoothing. The smooth function $\hat{f}(t)$ is computed at a given point and is a linear combination of local observations,

$$\hat{f}(t) = \sum_{i=1}^{n} S_j(t) f(t_j)$$

for some defined weight functions S_j. The most popular weight function are those defined by the Nadaraya-Watson estimator [48, 72] and is constructed using the weights

$$S_j(t) = \frac{\text{Kern}[(t_j - t)/h]}{\sum_r \text{Kern}[(t_r - t)/h]},$$

where Kern is a kernel function obeying Mercer's conditions and h is the bandwidth of the kernel. For more methods on constructing weights see the method of Gasser and Müller [16] and Ramsay and Silverman [52].

Roughness Penalty. Another popular way to smooth functional data is to add a roughness penalty to the least-squares fit. The way to quantify the notion of roughness is the square of the second derivative, $[D^2 f(t)]^2$ which is also known as the curvature at t. Therefore, a natural measure of a function's roughness is the integrated squared second derivative

$$\text{pen}(f) = \int [D^2 f(t)]^2 \, dt.$$

We then can define a compromise that trades of smoothness against data fit of a basis using the penalized least-squares fit,

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} (\mathbf{f} - \Theta\mathbf{b})^\mathsf{T} (\mathbf{f} - \Theta\mathbf{b}) + \lambda \, \text{pen}(\Theta\mathbf{b}),$$

and the parameter λ is the smoothing parameter and controls the amount of smoothing. Fig. 2.2 presents an example of smoothing the same data in Fig. 2.1 using the roughness penalty for different values of λ.

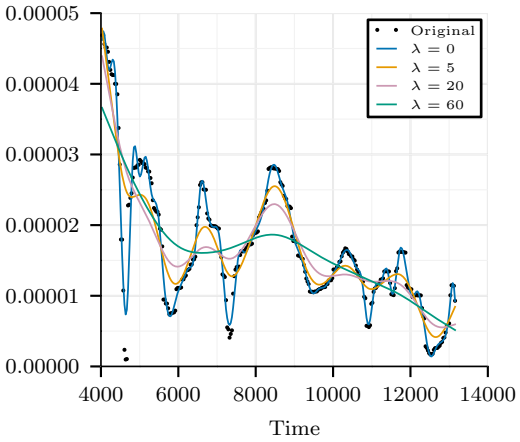


Figure 2.2: Example of smoothing of functional data by changing the amount of smoothing penalty.

2.1.3 Functional Principal Component Analysis

The motivation for functional principal component analysis (fPCA) is that the directions of high variance will contain more information then direction of low variance. Let $f_1, \cdots, f_n$ be a given set of functions, the optimization problem for fPCA can be written as

$$\min_{w_i} E\|f - \hat{f}\|^2 \tag{2.1.1}$$

where $\hat{f} = \mu_f + \sum_{i=1}^{n} \beta_i w_i(t)$ is the fPCA approximation of f with corresponding mean μ_f, $\beta_i = \int (f - \mu_f) w_i(t)\, dt$, and basis functions $\{w_i(t)\}$.

We then use the sample covariance function $\text{cov}(t_1, t_2)$ to form a sample covariance matrix K. Taking the SVD, $K = U\Sigma V^{\mathsf{T}}$ we can calculate the directions of principle variability in the given functions using the first $p \leq n$ columns of U. Moreover, we can calculate the observed principal coefficients as $\langle f_i, U_j \rangle$.

One can then use this framework to visualize the principal-geodesic paths. The basic idea is to compute a few points along geodesic path $\tau \mapsto \bar{f}(t) + \tau\sqrt{\Sigma_{jj}} U_j$ for $\tau \in \mathbb{R}$ in $\mathbb{L}^2$, where Σ_{jj} and U_j are the j^{th} singular value and column, respectively.

2.1.4 Functional Regression

In functional data analysis, regression modeling is where the function variables are used as predictors to estimate a scalar response variable. More precisely, let the predictor functions be

given by $\{f_i : [0,T] \to \mathbb{R},\ i = 1, 2, \ldots, n\}$ and the corresponding response variables be y_i. The standard functional linear regression model for this set of observations is

$$y_i = \alpha + \int_0^T f_i(t)\beta(t)\,dt + \epsilon_i, \quad i = 1, \ldots, n \tag{2.1.2}$$

where $\alpha \in \mathbb{R}$ is the intercept, $\beta(t)$ is the regression-coefficient function and $\epsilon_i \in \mathbb{R}$ are random errors. This model was first studied in [50] and [9]. The model parameters are usually estimated by minimizing the sum of squared errors (SSE),

$$\{\alpha^*, \beta^*(t)\} = \arg\min_{\alpha,\beta(t)} \sum_{i=1}^{n} |y_i - \alpha - \int_0^T f_i(t)\beta(t)\,dt|^2.$$

These values form maximum-likelihood estimators of parameters under the additive white-Gaussian noise model. One problem with this approach, is that for any finite n, since β is a full function there are infinitely many solutions for β without imposing any further restrictions; it is an element of an infinite-dimensional space while its specification for any n is finite dimensional. Ramsay [52] proposed two approaches to handle this issue: (1) Represent $\beta(t)$ using p basis functions in which p is kept large to allow desired variations of $\beta(t)$, and (2) add a roughness penalty term to the objective function (SSE) which selects a smooth solution by finding an optimal balance between the SSE and the roughness penalty. The basis can come from any of the basis functions described earlier, or fPCA [54].

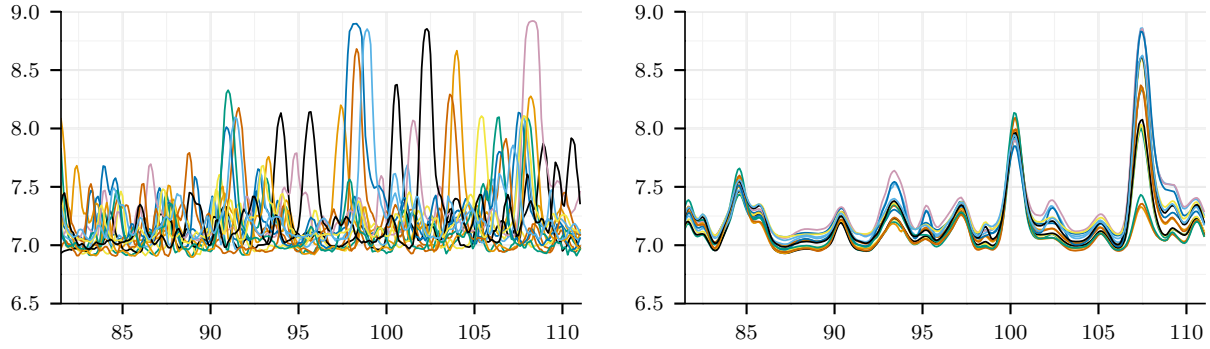
Current literature in functional linear regression is focused primarily on the estimation of the coefficient of $\beta(t)$ under a basis representation. For example, [10, 14, 21, 25] discuss estimation and/or inference of $\beta(t)$ for different cases for the standard functional linear model and the interpretation of $\beta(t)$. Focusing on prediction of the scalar response, [8] studied the estimation of $\int f_i(t)\beta(t)\,dt$. In some situations the response variable, y_i is categorical and the standard linear model will not suffice. James [23], extended the standard functional linear model to functional logistic regression to be able to handle such situations. Müller and Stadtmüller [47], extend the generalized model to contain a dimension reduction by using a truncated Karhunen-Loève expansion. Recently, [17] included variable selection to reduce the number of parameters in the generalized model.

2.2 Phase-Amplitude Separation

All of the methods in the previous section assume the data has no phase-variability or is aligned. For most data that is observed, this is not the case. For example Fig. 2.3a presents protein profiles

of five patients with Acute Myeloid Leukemia (AML) [38] which exhibit no alignment at all. These profiles should be aligned, but due to sensor and measurement error, this is not the case. Delaying the details till Section 2.2.2, we can align the profiles [70] for statistical analysis and the aligned profiles are presented in Fig. 2.3b. There exists a large set of literature on the registration or alignment of functional data, in part due to the work of Ramsay and Silverman [52], Kniep and Gasser [36], and Müller et al. [43, 65]. The main difference between them lies in the choice of the cost function used in the alignment process.

First, we will define some mathematical representation of warping. Let Γ be the set of warping functions. For any $f \in \mathcal{F}$ and $\gamma \in \Gamma$, the composition $f \circ \gamma$ denotes the time-warping of f by γ. In a pairwise alignment problem, the goal is to align any two functions f_1 and f_2. A majority of past methods use cost terms of the type $(\inf_{\gamma \in \Gamma} \|f_1 - f_2 \circ \gamma\|)$ to perform this alignment. Here $\|\cdot\|$ denotes the standard $\mathbb{L}^2$ norm.



(a) Original Data

(b) Aligned Data

Figure 2.3: Example of data with a) phase- and amplitude-variability and b) aligned data.

2.2.1 Previous Work

Liu and Müller [43] use warping functions that are convex combinations of functions of the type: $\gamma_i(t) = \left(\frac{\int_0^t |f_i^{(\nu)}(s)|^p ds}{\int_0^1 |f_i^{(\nu)}(s)|^p ds} \right)^{(1/p)}$, where ν and p are two parameters, with the recommended values being $\nu = 0$ and $p = 1$. The warped functions are then analyzed using standard statistical techniques under the Hilbert structure of square-integrable functions. Tang and Müller [65] then follow the previous work by aligning the functions using the area-under- the curve as the metric.

Ramsay and Silverman [52] proposed using warping functions of the structure $\gamma(t) = C_0 + C_1 \int_0^1 \exp(W(u))du$, where C_0 and C_1 are constants that are fixed by the requirement that $\gamma(t) = t$, and $W(u)$ controls the actual warping. The alignment is done by minimizing the second-eigenvalue of the functional analogue of the cross-product matrix $X^\mathsf{T}X$

$$T(f_1, f_2) = \begin{bmatrix} \int f_1(t)^2 dt & \int f_1(t) f_2 \circ \gamma(t) dt \\ \int f_2 \circ \gamma(t) f_1(t) dt & \int f_2 \circ \gamma(t)^2 dt \end{bmatrix},$$

where f_1 and f_2 are the two functions to be registered. Additionally, they apply regularization to the minimization problem to impose smoothness on the structure of $\gamma(t)$.

James [24] proposed an approach that calculates moments for each function and then aligns the functions by matching the moments. The motivation behind using moments is that they are intended to capture the locations of important features (e.g., maximums and minimums) of the functional data. The proposed warping functions in this case were linear and of the form $\gamma(t) = \alpha_i + \beta_i t$. Gervini and Gasser [19] uses a self-modeling framework where the warping functions are assumed to be linear combinations of q components, which are estimated from the data. In other words, a semi-parametric model is constructed for the functional data.

Recently, Sangalli et al. [56, 57] proposed a method that jointly clusters and aligns the functional data using a modified k-means approach. The alignment is done by finding warping functions of the linear form similar to [24] that maximize a similarity index. The similarity index is defined to be

$$\rho(f_1, f_2) = \frac{1}{d} \sum_{p=1}^{n} \frac{\int f'_{1p}(t) f'_{2p}(t) dt}{\sqrt{\int f'_{1p}(t)^2 dt} \sqrt{\int f'_{2p}(t)^2 dt}},$$

where $f_{ip}(t)$ is the pth time sample of function f_i. Geometrically speaking, the similarity index is the average of the cosines of the angles between the derivatives of the functions f_1 and f_2.

In the meantime several other communities, often outside statistics, have studied registration of functions in one or higher dimensions, e.g., in matching MRI images [2, 13, 64], shape analysis of curves [27, 35, 41, 45, 62, 76, 77], shape analysis of surfaces [39], etc. The problem of curve registration is especially relevant for phase-amplitude separation needed in functional data analysis, since the case for $\mathbb{R}^1$ is essentially that of real valued functions.

The majority of past methods in the statistical literature (e.g., [19, 24, 37, 43, 65]) study the problem of registration and comparisons of functions, either separately or jointly, by solving:

$$\inf_{\gamma \in \Gamma} \|f_1 - (f_2 \circ \gamma)\|. \tag{2.2.1}$$

The use of this quantity is problematic because it is not symmetric. The optimal alignment of f_1 to f_2 gives a different minimum, in general, when compared to the optimal alignment of f_2 to f_1. One can make Eqn. 2.2.1 symmetric by using double optimization, $\inf_{(\gamma_1,\gamma_2)\in\Gamma\times\Gamma} \|(f_1 \circ \gamma_1) - (f_2 \circ \gamma_2)\|$. However, this problem is degenerate, in the sense that the solution can be made infinitesimally small as long as the range(f_1) and range(f_2) intersect, even if the two functions f_1 and f_2 are quite different. Another way of ensuring symmetry is to add the two terms: $\inf_{\gamma\in\Gamma} \|f_1 - (f_2 \circ \gamma)\| + \|f_2 - (f_1 \circ \gamma)\|$. Although this expression is symmetric, it still does not lead to a proper distance between the space of functions.

The basic quantity in Eqn. 2.2.1 is also commonly used to form objective functions of the type:

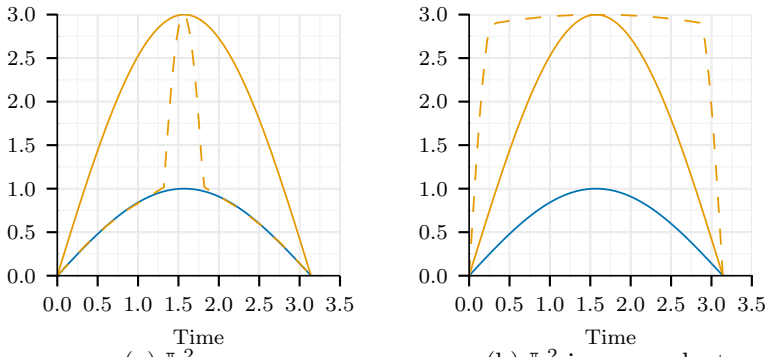
$$E_{\lambda,i}[\mu] = \inf_{\gamma_i\in\Gamma} \left(\|(f_i \circ \gamma_i) - \mu\|^2 + \lambda \mathcal{R}(\gamma_i) \right), \quad i = 1, 2, \ldots, n \ , \tag{2.2.2}$$

where $\mathcal{R}$ is a smoothness penalty on the γ_is to keep them close to $\gamma_{id}(t) = t$ [52]. The optimal γ_i^* are then used to align the f_is, followed by a cross-sectional analysis of the aligned functions. There are couple of issues here: 1) How is the parameter λ chosen for the regularization, this could vastly vary from problem to problem and 2) What should the mean, μ, be? It cannot be a mean of the f_is since the quantity in Eqn. 2.2.1 is not a *proper distance*. Specifically, the cost function does not have the properties of symmetry (optimal alignment of f to g is same as that of g to f), positive definiteness (the cost term between any two functions is nonnegative and it equals zero if and only if one can be perfectly aligned to the other), and the triangle inequality. Additionally, many past methods perform component separation and modeling in two distinct steps, under different metrics. Moreover, the group structure of Γ is seldom utilized and the construction of the warping functions usually does not result in a proper group.

Another large problem with the use of the $\mathbb{L}^2$ metric is known as the *pinching problem* [51]. Specifically, if we have two functions, f_1 and f_2 and the range(f_1) is entirely above the range(f_2) the $\mathbb{L}^2$ metric becomes degenerate and pinching of the warped function occurs. Using the $\mathbb{L}^2$ norm or inner product we can demonstrate the pinching effect for four cases:

1. $\sup_\gamma \|f \circ \gamma\|$
2. $\inf_\gamma \|f_1 - f_2 \circ \gamma\|$
3. $\sup_\gamma \langle f_1, f_2 \circ \gamma \rangle$
4. $\sum_{i=1}^n \inf_{\gamma_i} \|f_i \circ \gamma_i - \mu\|^2$

These cases are of importance as the $\mathbb{L}^2$ norm and inner product are commonly used throughout most of the published work in functional data analysis.



(a) $\mathbb{L}^2$ norm (b) $\mathbb{L}^2$ inner-product

Figure 2.4: Demonstration of pinching problems in $\mathcal{F}$ space under (a) the $\mathbb{L}^2$ distance and (b) the $\mathbb{L}^2$ norm.

In case 1) we desire to have the norm preserved under the warping of f, however this is not the case in $\mathcal{F}$ space as $\|f \circ \gamma\| \neq \|f\|$ as the $\|f \circ \gamma\|$ can be made as large as the range of f. Fig. 2.4 presents examples of cases 2) and 3) in Panels a and b, respectively. The functions $f_1 = \sin(t)$ and $f_2 = 3\sin(t)$, $t \in [0, \pi]$ are shown as the blue and orange curves, respectively. In Panel a we demonstrate finding the γ that minimizes the $\mathbb{L}^2$ norm between f_1 and f_2 and the dashed blue curve is the warped $f_2 \circ \gamma$ using the optimal γ. A similar effect is seen in Panel b where we demonstrate finding the γ that maximizes the $\mathbb{L}^2$ inner product between f_1 and f_2 and the dashed blue curve is the warped $f_2 \circ \gamma$ using the optimal γ. Both the $\mathbb{L}^2$ norm and inner-product become degenerate and the resulting functions are "pinched." In case 4) if the functions f_i have a point in common in their ranges then $\mu(t)$ will be a constant function and be degenerate. For example, if for all t_i we have $f_i(t_i) = \mu$, then we can find $\gamma_i(t) = t_i$ for all t which gives a cost function value of 0 and a degenerate solution. This is important in the multiple alignment problem as one seeks to find a $\mu(t)$ to align the set of functions, $\{f_i(t)\}$ to. One solution to the pinching is to place a penalty on the warping functions as in Eqn. 2.2.2, though the issue here is how is the penalty parameter λ chosen.

Recently, Srivastava et al. [31, 40, 63] adapted a shape-analysis approach that has been termed *elastic shape analysis* [27, 41, 62]. The basic idea in this method is to introduce a mathematical

representation; called the *square-root slope function* or SRSF (details in the next section) that improves functional alignment, and provides fundamental mathematical equalities that lead to a formal development of this topic. Moreover, the metric used in the alignment is a proper distance and avoids the pinching effects of the standard $\mathbb{L}^2$ metric in $\mathcal{F}$ space without the use of a penalty.

2.2.2 Phase and Amplitude Separation Using Elastic Analysis

In this section, we review the elastic method for functional data alignment. The details are presented in companion papers [40, 63]. This comprehensive framework addresses three important goals: (1) completely automated alignment of functions using nonlinear time warping, (2) separation of phase and amplitude components of functional data, and (3) derivation of individual phase and amplitude metrics for comparing and classifying functions. For a more comprehensive introduction to this theory, including asymptotic results and estimator convergences, we refer the reader to these papers as we will only present the algorithm here.

Let Γ be the set of boundary-preserving diffeomorphisms of the unit interval $[0,1]$: $\Gamma = \{\gamma : [0,1] \to [0,1] |\ \gamma(0) = 0,\ \gamma(1) = 1, \gamma \text{ is a diffeomorphism}\}$. As before elements of Γ play the role of warping functions. For any $f \in \mathcal{F}$ and $\gamma \in \Gamma$, the composition $f \circ \gamma$ denotes the time-warping of f by γ. With the composition operation, the set Γ is a group with the identity element $\gamma_{id}(t) = t$.

To overcome the problem of the alignment ($\inf_{\gamma\in\Gamma} \|f_1 - f_2 \circ \gamma\|$) not being symmetric nor positive definite Srivastava et al. [62] introduced a mathematical expression for representing a function.

This function, $q : [0,1] \to \mathbb{R}$, is called the *square-root slope function* or SRSF of f, and is defined in the following form:

$$q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}\ .$$

It can be shown that if the function f is absolutely continuous, then the resulting SRSF is square-integrable (see [55] for a proof). Thus, we will define $\mathbb{L}^2([0,1],\mathbb{R})$, or simply $\mathbb{L}^2$, to be the set of all SRSFs. For every $q \in \mathbb{L}^2$ and a fixed $t \in [0,1]$, the function f can be obtained precisely using the equation: $f(t) = f(0) + \int_0^t q(s)|q(s)|ds$, since $q(s)|q(s)| = \dot{f}(s)$. Therefore, the mapping from $\mathcal{F}$ to $\mathbb{L}^2 \times \mathbb{R}$, given by $f \mapsto (q, f(0))$ is a bijection [55]. The most important property of this framework is the following. If we warp a function f by γ, the SRSF of $f \circ \gamma$ is given by: $\tilde{q}(t) = (q,\gamma)(t) = q(\gamma(t))\sqrt{\dot{\gamma}(t)}$. With this expression it can be shown that for any $f_1, f_2 \in \mathcal{F}$ and $\gamma \in \Gamma$, we have

$$\|q_1 - q_2\| = \|(q_1,\gamma) - (q_2,\gamma)\|\ , \tag{2.2.3}$$

where q_1, q_2 are SRSFs of f_1, f_2, respectively. This is called the *isometry* property and it is central in suggesting a new cost term for pairwise registration of functions: $\inf_{\gamma\in\Gamma} \|q_1 - (q_2, \gamma)\|$. This equation suggests we can register (or align) the SRSFs of any two functions first and then map them back to $\mathcal{F}$ to obtain registered functions. The advantage of this choice is that it is symmetric, positive definite, and satisfies the triangle inequality. Technically, it forms a proper distance[1] on the quotient space $\mathbb{L}^2/\Gamma$. Moreover, this metric does not exhibit the pinching problem as does the $\mathbb{L}^2$ metric in $\mathcal{F}$ space. We mention that this cost function has a built-in regularization term and does not require any additional penalty term. Please refer to papers [40, 63] for more details. In case one wants to control the amount of warping or *elasticity* this can be done as described in [74].

The isometric property in Eqn. 2.2.3 leads to a distance between functions that is *invariant* to their random warpings:

Definition 1 (Amplitude or y-distance)**.** *For any two functions $f_1,\ f_2 \in \mathcal{F}$ and the corresponding SRSFs, $q_1, q_2 \in \mathbb{L}^2$, we define the amplitude or the y-distance d_a to be:*

$$d_a(f_1, f_2) = \inf_{\gamma\in\Gamma} \|q_1 - (q_2 \circ \gamma)\sqrt{\dot{\gamma}})\|. \tag{2.2.4}$$

It can be shown that for any $\gamma_1, \gamma_2 \in \Gamma$, we have $d_a(f_1 \circ \gamma_1, f_2 \circ \gamma_2) = d_a(f_1, f_2)$.

Optimization Over Γ. The minimization over Γ can be performed in many ways. In case Γ is represented by a parametric family, then one can use the parameter space to perform the estimation as [37]. However, since Γ is a nonlinear manifold, it is impossible to express it completely in a parametric vector space.

Remark: The set of warping functions Γ is not a closed set and, therefore, the maximizer of log-likelihood over γ_i may not be in Γ. From a technical perspective, this situation can be handled by enlarging the set of warping functions to include all nondecreasing, absolutely-continuous functions with the same boundary conditions ($\gamma(0) = 0$ and $\gamma(1) = 1$). This set, termed $\tilde{\Gamma}$ is a closed set and has the property that Γ is dense in $\tilde{\Gamma}$. The optimization over γ_i can now be performed in the larger set $\tilde{\Gamma}$. From a practical point of view, the change from Γ to $\tilde{\Gamma}$ is not significant since the former is dense in latter.

[1] We note that restriction of $\mathbb{L}^2$ metric to SRSFs of functions whose first derivative is strictly positive, e.g., cumulative distribution functions, is exactly the classical Fisher-Rao Riemannian metric used extensively in the statistics community [1, 11, 15, 30, 53].

In this dissertation we use the standard Dynamic Programming algorithm [3] to solve for an optimal γ. It should be noted that for any fixed partition of the interval $[0, 1]$, this algorithm provides the exact optimal γ that is restricted to the graph of this partition.

2.2.3 Karcher Mean and Function Alignment

In order to separate phase and amplitude variability in functional data, we need a notion of the mean of functions. Basically, we compute a mean function and in the process warp the given functions to match the mean function. Since we have a **proper** distance in d_a, in the sense that it is invariant to random warping, we can use that to define this mean. For a given collection of functions $f_1, f_2, \ldots, f_n$, let $q_1, q_2, \ldots, q_n$ denote their SRSFs, respectively. Define the Karcher mean of the given function as a local minimum of the following cost functions:

$$\mu_f = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} d_a(f, f_i)^2 \tag{2.2.5}$$

$$\mu_q = \arg\min_{q \in \mathbb{L}^2} \sum_{i=1}^{n} \left(\inf_{\gamma_i \in \Gamma} \|q - (q_i, \gamma_i)\|^2 \right) . \tag{2.2.6}$$

(This Karcher mean has also been called by other names such as the Frechet mean, intrinsic mean or the centroid.) These are two equivalent formulations, one in the function space $\mathcal{F}$ and other in the SRSF space $\mathbb{L}^2$, i.e., $\mu_q = \text{sign}(\dot{\mu}_f)\sqrt{|\dot{\mu}_f|}$. Note that if μ_f is a minimizer of the cost function, then so is $\mu_f \circ \gamma$ for any $\gamma \in \Gamma$ since d_a is invariant to random warpings of its input variables. So, we have an extra degree of freedom in choosing an arbitrary element of the set $\{\mu_f \circ \gamma | \gamma \in \Gamma\}$. To make this choice unique, we can define a special element of this class as follows. Let $\{\gamma_i^*\}$ denote the set of optimal warping functions, one for each i, in Eqn. 2.2.6. Then, we can choose the μ_f to that element of its class such that the mean of $\{\gamma_i^*\}$, denoted by γ_μ, is γ_{id}, the identity element. (The notion of the mean of warping functions and its computation are described later in Chapter 3 and summarized in Algorithm 3.1). The algorithm for computing the Karcher mean μ_f of SRSFs is given in Algorithm 2.1, where the iterative update in Steps 2-4 is based on the gradient of the cost function given in Eqn. 2.2.6.

This procedure results in three items:

1. μ_q, preferred element of the Karcher mean class $\{(\mu_q, \gamma) | \gamma \in \Gamma\}$,
2. $\{\tilde{q}_i\}$, the set of aligned SRSFs,

Algorithm 2.1 Phase-Amplitude Separation

1: Compute SRSFs $q_1, q_2, \ldots, q_n$ of the given $f_1, f_2, \ldots, f_n$ and select $\mu = q_i$, where $i = \arg\min_{1 \leq i \leq n} ||q_i - \frac{1}{n}\sum_{j=1}^{n} q_j||$.

2: For each q_i find the γ_i^* such that $\gamma_i^* = \arg\min_{\gamma \in \Gamma} \left(\|\mu - (q_i \circ \gamma)\sqrt{\dot{\gamma}}\| \right)$. The solution to this optimization comes from the dynamic programming algorithm.

3: Compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$.

4: If the increment $\|\frac{1}{n}\sum_{i=1}^{n} \tilde{q}_i - \mu\|$ is small, then stop. Else, update the mean using $\mu \mapsto \frac{1}{n}\sum_{i=1}^{n} \tilde{q}_i$ and return to step 2.

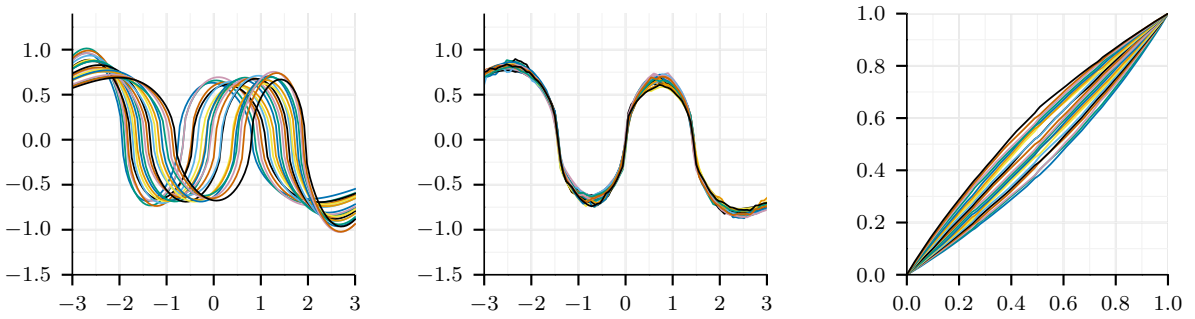
5: The function μ represents a whole equivalence class of solutions and now we select the preferred element μ_q of that orbit:

   1. Compute the mean γ_μ of all $\{\gamma_i^*\}$ (using Algorithm 3.1). Then compute $\mu_q = (\mu \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma}_\mu^{-1}}$.
   2. Update $\gamma_i^* \mapsto \gamma_i^* \circ \gamma_\mu^{-1}$. Then compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$ and aligned functions using $\tilde{f}_i \mapsto f_i \circ \gamma_i^*$

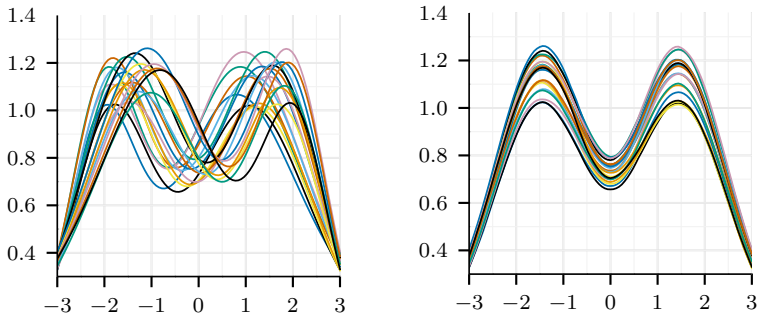
3. $\{\tilde{f}_i\}$, the set of aligned functions, and
4. $\{\gamma_i^*\}$, the set of optimal warping functions.

To illustrate this method, we run the algorithm on the data previously used in [37]. The individual functions are given by: $y_i(t) = z_{i,1}e^{-(t-1.5)^2/2} + z_{i,2}e^{-(t+1.5)^2/2}$, $t \in [-3, 3]$, $i = 1, 2, \ldots, 21$, where $z_{i,1}$ and $z_{i,2}$ are *i.i.d.* $\mathcal{N}(1, (0.25)^2)$. (Note that although the elastic framework was developed for functions on $[0, 1]$, it can easily be adapted to an arbitrary interval). Each of these functions is then warped according to: $\gamma_i(t) = 6\left(\frac{e^{a_i(t+3)/6}-1}{e^{a_i}-1}\right) - 3$ if $a_i \neq 0$, otherwise $\gamma_i = \gamma_{id}$ ($\gamma_{id}(t) = t$ is the identity warping). Here a_i are equally spaced between -1 and 1, and the observed functions are computed using $f_i(t) = (y_i \circ \gamma_i)(t)$. A set of 21 such functions forms the original data and is shown in Panel d of Fig. 2.5 with corresponding SRSFs in Panel a. Panel b presents the resulting aligned SRSFs using our method $\{\tilde{q}_i\}$ and Panel c plots the corresponding warping functions $\{\gamma_i^*\}$. The corresponding aligned functions $\{\tilde{f}_i\}$ is shown in Panel e. It is apparent that the plot of $\{\tilde{f}_i\}$ shows a tighter alignment of functions with sharper peaks and valleys, and thinner bands around the mean. This indicates that the effects of warping generated by the γ_is has been completely removed and only the randomness from the y_is remain.

We also compare the performance of Algorithm 2.1 with some published methods including: the moment based matching (MBM) method of [24] and the minimum second-eigenvalue (MSE) method



(a) Original SRSFs $\{q_i\}$ (b) Warped Data SRSFs $\{\tilde{q}_i\}$ (c) Warping Functions $\{\gamma_i^*\}$

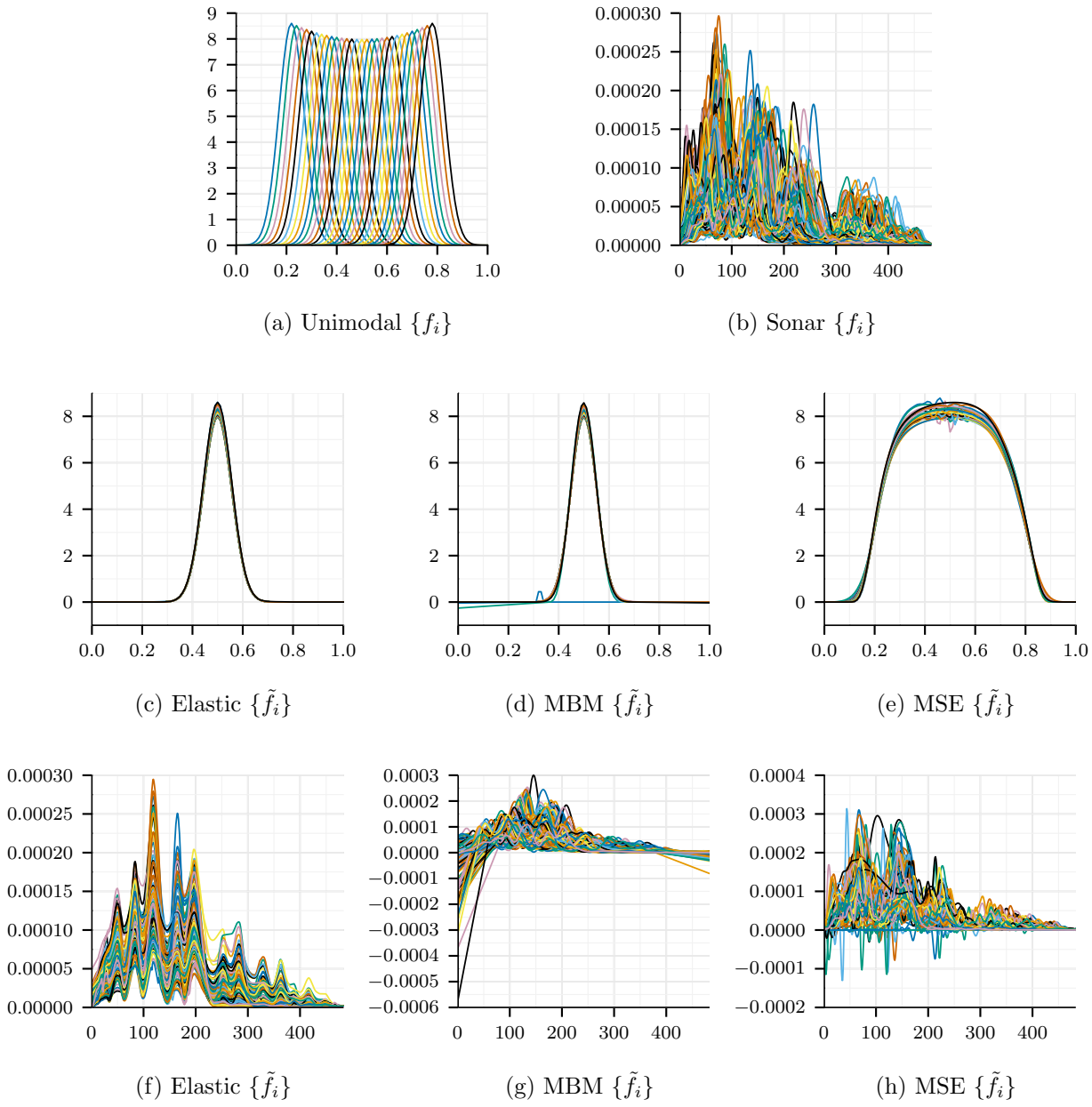


(d) Original Data $\{f_i\}$ (e) Warped Data $\{\tilde{f}_i\}$

Figure 2.5: Alignment of the simulated data set using Algorithm 2.1.

of [52] on a more difficult simulated data and a real SONAR data set. The original simulated data are shown in Figs. 2.6a and the data consists of 39 unimodal functions which have been warped with equally spaced centers along the x-axis and, have slight variation in peak-heights along the y-axis. Figs. 2.6 c, d, and e present the alignment results for our elastic method, the MBM method, and the MSE method, respectively. The original SONAR data are shown in Fig. 2.6b and the data consists of 131 measured SONAR signals that contain measurement ambiguity. Fig. 2.6 f, g, and h present the alignment results for our elastic method, the MBM method, and the MSE method, respectively.

For the simulated data the elastic method performs the best, while the MBM method performs fairly well with a little higher standard deviation. The MBM method and the MSE method both have a few numerical issues that lead to blips in the functions. For the SONAR data, only the elastic method performs well; as MBM and MSE methods fail to align the data. We can also



(a) Unimodal $\{f_i\}$ (b) Sonar $\{f_i\}$

(c) Elastic $\{\tilde{f}_i\}$ (d) MBM $\{\tilde{f}_i\}$ (e) MSE $\{\tilde{f}_i\}$

(f) Elastic $\{\tilde{f}_i\}$ (g) MBM $\{\tilde{f}_i\}$ (h) MSE $\{\tilde{f}_i\}$

Figure 2.6: Comparison of alignment algorithms on a difficult unimodal data set (second row) and a real SONAR data set (bottom row).

quantify the alignment performance using the decrease in the cumulative cross-sectional variance of the aligned functions. For any functional dataset $\{g_i(t), i = 1, 2, \ldots, n, t \in [0, 1]\}$, let

$$\text{Var}(\{g_i\}) = \frac{1}{n-1} \int_0^1 \sum_{i=1}^{n} \left(g_i(t) - \frac{1}{n} \sum_{i=1}^{n} g_i(t) \right)^2 dt \ ,$$

denote the cumulative cross-sectional variance in the given data. With this notation, we define:

$$\begin{aligned} \text{Original Variance} &= \text{Var}(\{f_i\}) \\ \text{Amplitude Variance} &= \text{Var}(\{\tilde{f}_i\}) \\ \text{Phase Variance} &= \text{Var}(\{\mu_f \circ \gamma_i\}). \end{aligned}$$

The phase- and amplitude-variances for the different alignment algorithms shown in Fig. 2.6 are listed below in Table 2.1; with the simulated unimodal data on the top two rows and the SONAR data on the bottom two rows: Based on its superior performance and theoretical advantages, we

Table 2.1: The comparison of the amplitude variance and phase variance for different alignment algorithms on the Unimodal and SONAR data sets.

Unimodal				
Component	Original Variance	Elastic Method	MBM	MSE
Amplitude-variance	4.33	**0.004**	0.23	.02
Phase-variance	0	4.65	4.31	4.54
Sonar				
Amplitude-variance	2.89e-5	**1.53e-5**	3.02e-5	2.42e-5
Phase-variance	0	1.48e-5	1.30e-5	1.36e-5

choose the elastic method for separating the phase and amplitude components. For additional experiments and asymptotic analysis of this method, please refer to [40, 63]. Recently, Cheng [12] has extended the elastic work using a Bayesian approach where they use a Dirichlet prior on the warping functions and a Markov chain Monte Carlo algorithm for finding the Karcher mean.

2.2.4 Functional Component Analysis with Alignment

Most of the current statistical papers that use some form of component analysis or statistical analysis on the function data is done post alignment. Specifically, the alignment is done *a priori* and both the analysis and alignment are done using a separate cost function. Sangalli et al. [56, 57]

performs k-means after alignment. Similarly, James [24] follows his MBM with standard-PCA and all of the methods demonstrated by Ramsay and Silverman in [52] assume the data is already aligned.

Instead of doing the procedures separately, the idea should be that of solving the alignment problem and the component analysis problem under the same cost function. The basic idea is to avoid treating warping as a *pre-processing* step; where the individual functions are warped according to an objective function that is different from the metric used to compare them. We feel the criteria for registration and comparison not only to be consistent, but actually to be the same. In other words, the domain warping of functions should be an integral part of the functional analysis.

Recently, Kniep and Ramsay [37] presented a joint cost function for performing alignment and performing functional principal component analysis (fPCA). The cost function was constructed such that the functions were aligned to the principal components and therefore the principal components were extracted as the warping functions are calculated. However, this method suffers from the same issues addressed in Section 2.2.2 where the cost function is the standard $\mathbb{L}^2$ in $\mathcal{F}$ space and is not a proper distance. Moreover, the selection of the mean is somewhat arbitrary given the true mean cannot be calculated.

Zhou and Torre [78] developed a method of combining canonical correlation analysis (CCA) and alignment, where their alignment is done using Dynamic Time Warping (DTW) [49]. Using the $\mathbb{L}^2$ cost function of CCA, they compute the canonical coordinates while calculating the alignment matrices of DTW. This approach has some pitfalls for functional data as they assume their data is zero-mean. One cannot always assume zero-mean warped data. Specifically, if a data is warped by $\{\gamma_i\}$ and the mean is not zero, the true mean is not known and the data cannot be centered. The cost function they use also suffers the same problems as [37] since it is the standard $\mathbb{L}^2$ in $\mathcal{F}$ space as it is not a proper distance. Additionally, their cost function is of the type $\inf_{(\gamma_1,\gamma_2)\in\Gamma\times\Gamma} \|(f_1 \circ \gamma_1) - (f_2 \circ \gamma_2)\|$ which is degenerate as explained in Section 2.2.2.

2.2.5 Functional Linear Regression with Alignment

As with functional component analysis most of the published regression methods ignore the phase-variability or pre-align the data. If the data is pre-aligned using an alignment method (e.g., [63], [52]) the underlying structure that separates classes in for instance a logistic regression model may be distorted. Recently, [18] has proposed a functional linear regression model that

includes phase-variability in the model. It uses a random-effect simultaneous linear model on the warping parameters and the principal component scores (of aligned predictor functions). However, the method involves landmark representations that limits its use in general regression models, particularly when the response is a categorical variable.

Therefore, we propose using the SRSF framework, motivated by benefits as explained previously, to construct cost functions for component analysis and regression models that are elastic. In other words, we can construct cost functions that perform the modeling and align the functions, while maintaining a proper metric.

CHAPTER 3

MODELING PHASE AND AMPLITUDE COMPONENTS

This chapter discusses the problem of modeling and classifying functional data after the phase and amplitude have been separated using the method described in Chapter 2.2.2. After the separation of phase- and amplitude-components, we will define two types of distances. One is a y- distance; defined to measure amplitude differences between any two functions (and independent of their phase-variability), computed as the $\mathbb{L}^2$ distance between the SRSFs of the aligned functions. The other is an x-distance, or the distance on their phase components, that measures the amount of warping needed to align the functions. We will show that either of these distances provides useful measures for computing summary statistics, performing fPCA, and discriminating between function classes. The main contribution of this chapter is a modeling framework to characterize functional data using phase and amplitude separation. The basic steps in this procedure are: 1) Align the original functional data to obtain the aligned functions (describing amplitude variability) and the warping functions (describing phase-variability); 2) Estimate the sample means and covariance on the phase and amplitude, respectively. This step uses a nonlinear transformation on the data to enable use of $\mathbb{L}^2$ norm (and cross-sectional computations) for generating summary statistics (see Section 3.1); 3) Based on the estimated summary statistics, perform fPCA on the phase and amplitude, respectively; 4) Model the original data by using joint Gaussian or non-parametric models on both fPCA representations; 5) As a direct application, the model can be used to perform likelihood-based classification of functional data. We will illustrate this application using several data sets which include a simulated data set, a signature data set from [75], an iPhone action data set from [44], and a SONAR data set. Additional results are presented in companion papers [67–70].

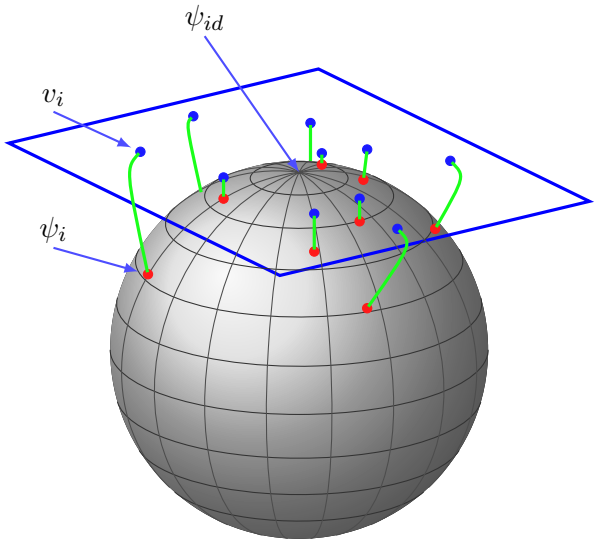


Figure 3.1: Depiction of the SRSF space of warping functions as a sphere and a tangent space at identity ψ_{id}.

3.1 Phase-Variability: Analysis of Warping Functions

First, we would like to study the phase-variability of the given functions, available to us in the form of the warping functions $\{\gamma_i^*\}$ resulting from Algorithm 2.1. An explicit statistical modeling of the warping functions can be of interest since they represent the phase-variability of the original data. As mentioned earlier, the space of warping functions, Γ, is a nonlinear manifold and cannot be treated as a Hilbert space directly. Therefore, we will use tools from differential geometry to be able to perform statistical analysis and modeling of the warping functions. This framework has been used previously, but in different application areas, e.g., modeling parameterizations of curves [60] and studies of execution rates of human activities in videos [71]. It also relates to the square-root representation of probability densities introduced by [4].

Let $\gamma_1, \gamma_2, \ldots, \gamma_n \in \Gamma$ be a set of observed warping functions. Our goal is to develop a probability model on Γ that can be estimated from the data directly. There are two problems in doing this in a standard way: (1) Γ is a nonlinear manifold, and (2) it is infinite dimensional. The issue of nonlinearity is handled using a convenient transformation which coincidentally is similar to the definition of SRSF, and the issue of infinite dimensionality is handled using dimension reduction, e.g., fPCA, which we will call *horizontal* fPCA. We are going to represent an element $\gamma \in \Gamma$ by

the square-root of its derivative $\psi = \sqrt{\dot{\gamma}}$. Note that this is the same as the SRSF defined earlier for f_is and takes this form since $\dot{\gamma} > 0$. The identity map γ_{id} maps to a constant function with value $\psi_{id}(t) = 1$. Since $\gamma(0) = 0$, the mapping from γ to ψ is a bijection and one can reconstruct γ from ψ using $\gamma(t) = \int_0^t \psi(s)^2 ds$. An important advantage of this transformation is that since $\|\psi\|^2 = \int_0^1 \psi(t)^2 dt = \int_0^1 \dot{\gamma}(t) dt = \gamma(1) - \gamma(0) = 1$, the set of all such ψs is a Hilbert sphere $\mathbb{S}_\infty$, a unit sphere in the Hilbert space $\mathbb{L}^2$. In other words, the square-root representation simplifies the complicated geometry of Γ to a unit sphere. The distance between any two warping functions is exactly the arc-length between their corresponding SRSFs on the unit sphere $\mathbb{S}_\infty$:

$$d_p(\gamma_1, \gamma_2) = d_\psi(\psi_1, \psi_2) \equiv \cos^{-1}\left(\int_0^1 \psi_1(t)\psi_2(t)dt\right) . \tag{3.1.1}$$

Fig. 3.1 shows an illustration of the SRSF space of warping functions as a unit sphere.

The definition of a distance on $\mathbb{S}_\infty$ helps define a Karcher mean of sample points on $\mathbb{S}_\infty$.

Definition 2. *For a given set of points $\psi_1, \psi_2, \ldots, \psi_n \in \mathbb{S}_\infty$, their Karcher mean in $\mathbb{S}_\infty$ is defined to be a local minimum of the cost function $\psi \mapsto \sum_{i=1}^n d_\psi(\psi, \psi_i)^2$.*

Now we can define the Karcher mean of a set of warping functions using the Karcher mean in $\mathbb{S}_\infty$. For a given set of warping functions $\gamma_1, \gamma_2, \ldots, \gamma_n \in \Gamma$, their Karcher mean in Γ is $\bar{\gamma}(t) \equiv \int_0^t \mu_\psi(s)^2 ds$ where μ_ψ is the Karcher mean of $\sqrt{\dot{\gamma}_1}$, $\sqrt{\dot{\gamma}_2}$, $\ldots$, $\sqrt{\dot{\gamma}_n}$ in $\mathbb{S}_\infty$. The search for this minimum is performed using Algorithm 3.1:

Algorithm 3.1 Karcher Mean of Warping Functions

1: Let $\psi_i = \sqrt{\dot{\gamma}_i}$ be the SRSFs for the given warping functions. Initialize μ_ψ to be one of the ψ_is or simply $w/\|w\|$, where $w = \frac{1}{n}\sum_{i=1}^n \psi_i$.
2: **while** $\|\bar{v}\|$ is small **do**
3: **for** i = 1:n **do**
4: Compute the shooting vector $v_i = \frac{\theta_i}{\sin(\theta_i)}(\psi_i - \cos(\theta_i)\mu_\psi)$, $\theta_i = \cos^{-1}(\langle \mu_\psi, \psi \rangle)$. By definition, each of these $v_i \in T_{\mu_\psi}(\mathbb{S}_\infty)$.
5: **end for**
6: Compute the average $\bar{v} = \frac{1}{n}\sum_{i=1}^n v_i \in T_{\mu_\psi}(\mathbb{S}_\infty)$.
7: Update $\mu_\psi \mapsto \cos(\epsilon\|\bar{v}\|)\mu_\psi + \sin(\epsilon\|\bar{v}\|)\frac{\bar{v}}{\|\bar{v}\|}$, for a small step size $\epsilon > 0$.
8: **end while**
9: Compute the mean warping function using $\bar{\gamma}(t) = \int_0^t \mu_\psi(s)^2 ds$.

Since $\mathbb{S}_\infty$ is a nonlinear space (a sphere), one cannot perform principal component analysis on it directly. Instead, we choose a vector space tangent to the space at a certain fixed point, for analysis. The tangent space at any point $\psi \in \mathbb{S}_\infty$ is given by: $T_\psi(\mathbb{S}_\infty) = \{v \in \mathbb{L}^2 | \int_0^1 v(t)\psi(t)dt = 0\}$. In the following, we will use the tangent space at μ_ψ to perform analysis. Note that the outcomes of Algorithm 3.1 include the Karcher mean μ_ψ and the tangent vectors $\{v_i\} \in T_{\mu_\psi}(\mathbb{S}_\infty)$. These tangent vectors, also called the *shooting vectors*, are the mappings of ψ_is into the tangent space $T_{\mu_\psi}(\mathbb{S}_\infty)$, as depicted in Fig. 3.1. In this tangent space we can define a sample covariance function: $(t_1, t_2) \mapsto \frac{1}{n-1}\sum_{i=1}^n v_i(t_1)v_i(t_2)$. In practice, this covariance is computed using a finite number of points, say T, on these functions. One obtains a $T \times T$ sample covariance matrix instead, denoted by K_ψ. The singular value decomposition (SVD) of $K_\psi = U_\psi \Sigma_\psi V_\psi^\mathsf{T}$ provides the estimated principal components of $\{\psi_i\}$: the principal directions $U_{\psi,j}$ and the observed principal coefficients $\langle v_i, U_{\psi,j}\rangle$. This analysis on $\mathbb{S}_\infty$ is similar to the ideas presented in [61], although one can also use the idea of principal nested sphere for this analysis [28].

As an example, we compute the Karcher mean of a set of random warping functions. These warping functions are shown in the left panel of Fig. 3.2 and their Karcher mean is shown in the second panel. Using the $\{v_i\}$'s that result from Algorithm 3.1, we form the covariance matrix K_ψ and take its SVD. The first three columns of U_ψ are used to visualize the principal geodesic paths in the third, fourth, and fifth panels.

3.2 Amplitude Variability: Analysis of Aligned Functions

Once the given observed SRSFs have been aligned using Algorithm 2.1, they can be statistically analyzed in a standard way, (in $\mathbb{L}^2$) using cross-sectional computations in the SRSF space. This is based on the fact that d_a (Eqn. 2.2.4) is the $\mathbb{L}^2$ distance between the aligned SRSFs. For example, one can compute the principal components for the purpose of dimension reduction and statistical modeling using fPCA. Since we are focused on the amplitude-variability in this section, we will call this analysis *vertical fPCA*.

Let $f_1, \cdots, f_n$ be a given set of functions, and $q_1, \cdots, q_n$ be the corresponding SRSFs, μ_q is their Karcher Mean, and let $\tilde{q}_i$s be the corresponding aligned SRSFs using Algorithm 2.1. In performing vertical fPCA, one should not forget about the variability associated with the initial values, i.e., $\{f_i(0)\}$, of the given functions. Since representing functions by their SRSFs ignores this initial

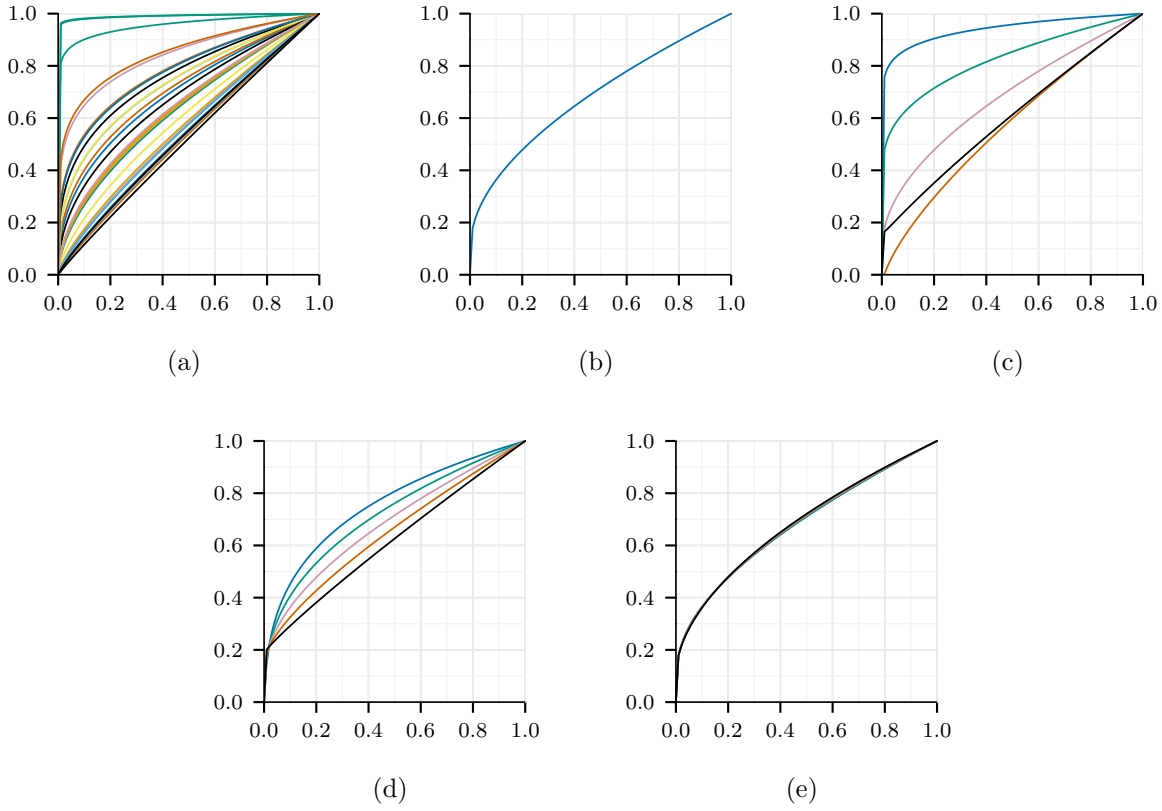


Figure 3.2: From left to right: (a) the observed warping functions, (b) their Karcher mean, (c) the first principal direction, (d) second principal direction, and (e) third principal direction of the observed data.

value, this variable is treated separately. That is, a functional variable f is analyzed using the pair $(q, f(0))$ rather than just q. This way, the mapping from the function space $\mathcal{F}$ to $\mathbb{L}^2 \times \mathbb{R}$ is a bijection. In practice, where q is represented using a finite partition of $[0, 1]$, say with cardinality T, the combined vector $h_i = [q_i \quad f_i(0)]$ simply has dimension $(T + 1)$ for fPCA. We can define a sample covariance operator for the aligned combined vector $\tilde{h}_i = [\tilde{q}_i \quad f_i(0)]$ as

$$K_h = \frac{1}{n-1} \sum_{i=1}^{n} E[(\tilde{h}_i - \mu_h)(\tilde{h}_i - \mu_h)^\mathsf{T}] \in \mathbb{R}^{(T+1)\times(T+1)} \ , \tag{3.2.1}$$

where $\mu_h = [\mu_q \quad \bar{f}(0)]$. Taking the SVD, $K_h = U_h \Sigma_h V_h^\mathsf{T}$ we can calculate the directions of principle variability in the given SRSFs using the first $p \leq n$ columns of U_h and can be converted back to the function space $\mathcal{F}$, via integration, for finding the principal components of the original functional data. Moreover, we can calculate the observed principal coefficients as $\left\langle \tilde{h}_i, U_{h,j} \right\rangle$.

One can then use this framework to visualize the vertical principal-geodesic paths. The basic idea is to compute a few points along geodesic path $\tau \mapsto \mu_h + \tau\sqrt{\Sigma_{h,jj}}U_{h,j}$ for $\tau \in \mathbb{R}$ in $\mathbb{L}^2$, where $\Sigma_{h,jj}$ and $U_{h,j}$ are the j^{th} singular value and column, respectively. Then, obtain principle paths in the function space $\mathcal{F}$ by integration as described earlier. Figure 3.3 shows the results of vertical fPCA on the simulated data set from Fig. 2.5. It plots the vertical principal-geodesic paths of the SRSFs, $q_{\tau,j}$ for $\tau = -2, -1, 0, 1, 2$ and $j = 1, 2, 3$ and the vertical principal-geodesic paths in function space. The first 3 singular values for the data are: 0.0481, 0.0307, and 0.0055 with the rest being negligibly small. The first principal direction corresponds to the height variation of the second peak while the second principal component captures the height variation of the first peak. The third principal direction has negligible variability.

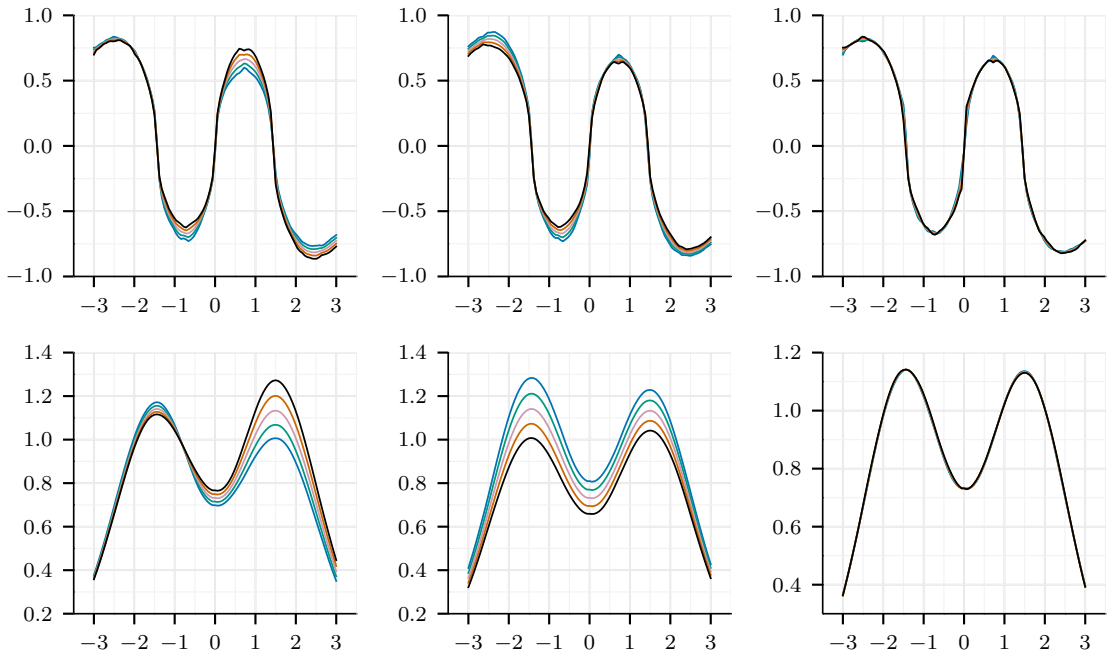


Figure 3.3: Vertical fPCA of aligned functions in simulated data set of Fig. 2.5. The first row shows the main three principal directions in SRSF space and the second row shows the main three principal directions in function space.

3.3 Modeling of Phase and Amplitude Components

To develop statistical models for capturing the phase and amplitude variability, there are several possibilities. Once we have obtained the fPCA coefficients for these components we can impose

probability on the coefficients and induce a distribution on the function space $\mathcal{F}$. Here we explore two possibilities: a joint Gaussian model and a non-parametric model.

Let $c = (c_1, \ldots, c_{k_1})$ and $z = (z_1, \ldots, z_{k_2})$ be the dominant principal coefficients of the amplitude- and phase-components, respectively, as described in the previous two sections. Recall that $c_j = \left\langle \tilde{h}, U_{h,j} \right\rangle$ and $z_j = \langle v, U_{\psi,j} \rangle$. We can reconstruct the amplitude component using $q = \mu_q + \sum_{j=1}^{k_1} c_j U_{h,j}$ and $f^s(t) = f^s(0) + \int_0^t q(s)|q(s)|ds$. Here, $f^s(0)$ is a random initial value. Similarly, we can reconstruct the phase component (a warping function) using $v = \sum_{j=1}^{k_2} z_j U_{\psi,j}$ and then using $\psi = \cos(\|v\|)\mu_\psi + \sin(\|v\|)\frac{v}{\|v\|}$, and $\gamma^s(t) = \int_0^t \psi(s)^2 ds$. Combining the two random quantities, we obtain a random function $f^s \circ \gamma^s$.

3.3.1 Gaussian Models on fPCA Coefficients

In this setup the model specification reduces to the choice of models for $f^s(0)$, c, and z. We are going to model them as multivariate normal random variables. The mean of $f^s(0)$ is $\bar{f}(0)$ while the means of c and z are zero vectors. Their joint covariance matrix is of the type: $\begin{bmatrix} \sigma_0^2 & L_1 & L_2 \\ L_1^\intercal & \Sigma_h & S \\ L_2^\intercal & S & \Sigma_\psi \end{bmatrix} \in \mathbb{R}^{(k_1+k_2+1)\times(k_1+k_2+1)}$. Here, $L_1 \in \mathbb{R}^{1\times k_1}$ captures the covariance between $f(0)$ and c, $L_2 \in \mathbb{R}^{1\times k_2}$ between $f(0)$ and z, and $S \in \mathbb{R}^{k_1\times k_2}$ between c and z. As discussed in the previous sections $\Sigma_h \in \mathbb{R}^{k_1\times k_1}$ and $\Sigma_\psi \in \mathbb{R}^{k_2\times k_2}$ are diagonal matrices and are estimated directly from the data. We will call this resulting probability model on the fPCA coefficients as p_{Gauss}.

3.3.2 Non-parametric Models on fPCA Coefficients

An alternative to the Gaussian assumption made above is the use of kernel density estimation [58], where the density of $f^s(0)$, each of the k_1 components of c, and the k_2 components of z can be estimated using

$$p_{ker}(x) = \frac{1}{nb}\sum_{i=1}^{n} \mathcal{K}\left(\frac{x - x_i}{b}\right)$$

where $\mathcal{K}(\cdot)$ is the smoothing kernel, which is a symmetric function that integrates to 1, and $b > 0$ is the smoothing parameter or bandwidth. A range of kernel functions can be used, but a common choice is the Gaussian kernel.

3.4 Modeling Results

We will now evaluate the models introduced in the previous section using random sampling. We will first estimate the means and the covariances from the given data, estimate the model parameters, then generate random samples based on the estimated models. We demonstrate results on two simulated data sets used in Figs. 2.5 and 2.6 and one real data set being the Berkeley growth data[1]. For the first simulated data set, shown in Fig. 2.5, we randomly generate 35 functions from the amplitude model and 35 domain-warping functions from the phase model, then combine them to generate random functions. The corresponding results are shown in Fig. 3.4, where the first panel is a set of random warping functions, the second panel is a set of corresponding amplitude functions, and the third panel shows their compositions. Comparing them with the original datasets (Fig. 2.5) we conclude that the random samples are very similar to the original data; at least under a visual inspection. The proposed models are successful in capturing the variability in the given data. Furthermore, if we compare these sampling results to the fPCA-based Gaussian model directly on f (without separating the phase and amplitude components) in the last panel of Fig. 3.4, we notice that our model is more consistent with the original data. A good portion of the samples from the non-separated model only contain three peaks or have a higher variation than the original data, with some barely representing the original data.

For the second simulated data set we use the data shown in Fig. 2.6 and perform vertical and horizontal fPCA. As before, we randomly generate 35 functions from the amplitude model and 35 domain-warping functions from the phase model and then combine them to generate random functions. The corresponding results are shown row of Fig. 3.5, where the first panel is a set of random warping functions, the second panel is a set of corresponding amplitude functions, and the last panel shows their compositions. Comparing them with the original data in Fig. 2.6 we conclude that the random samples are very similar to the original data. Under visual inspection, the proposed models are successful in capturing the variability in the given data. In this example performing fPCA directly on the function space does not correctly capture the data and fails to generate any single unimodal function shown in the last panel.

For the Berkley growth data, we again develop our phase and amplitude models, then randomly generate 35 functions from the amplitude model and 35 domain- warping functions from the phase

[1] http://www.psych.mcgill.ca/faculty/ramsay/datasets.html

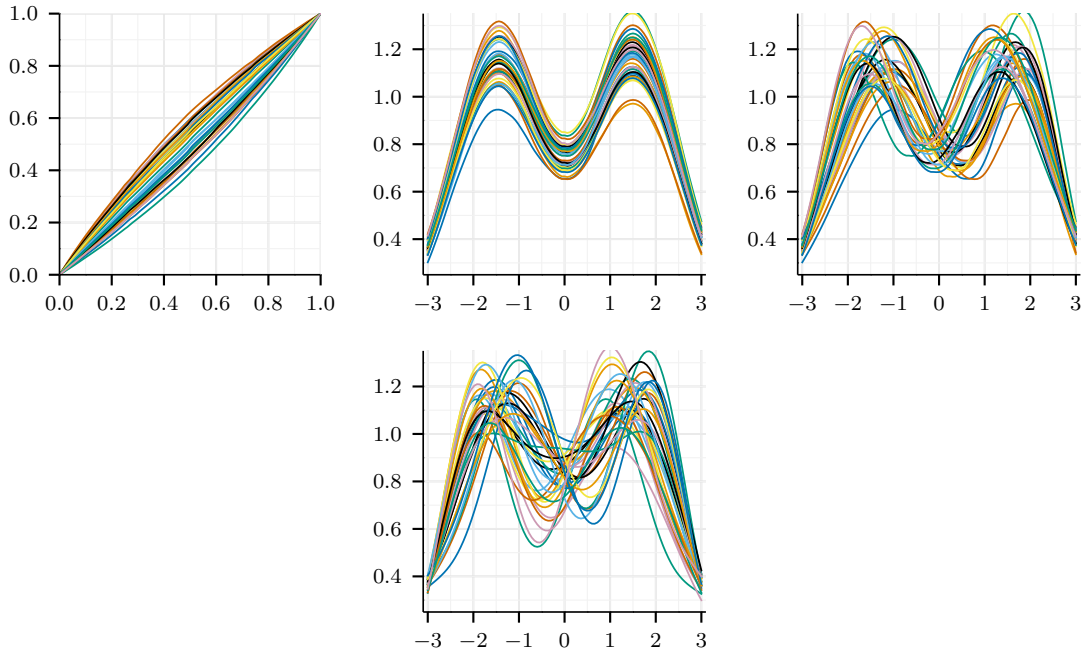


Figure 3.4: Random samples from jointly Gaussian models on fPCA coefficients of γ^s (left) and f^s (middle), and their combinations $f^s \circ \gamma^s$ (right) for Simulated Data 1. The last plot are random samples if a Gaussian model is imposed on f directly without any phase and amplitude separation.

model. We then compose them to generate random functions. The corresponding results are shown row of Fig. 3.6, where the first panel is a set of random warping functions, the second panel is a set of corresponding amplitude functions, and the last panel shows their compositions. Comparing them with the original data set in the last panel we conclude that the random samples are similar to the original data; and at least under a visual inspection, the proposed models are successful in capturing the variability in the given data.

3.5 Classification using Phase and Amplitude Models

An important use of statistical models of functional data is in classification of future data into pre-determined categories. Since we have developed models for both amplitude and phase, one or both can be used for classification and analyzed for their performance. Here we use a classical setup: a part of the data is used for training and estimation of model parameters, while the remaining part is used for testing. This partition is often random and repeated many times to obtain an average classification performance.

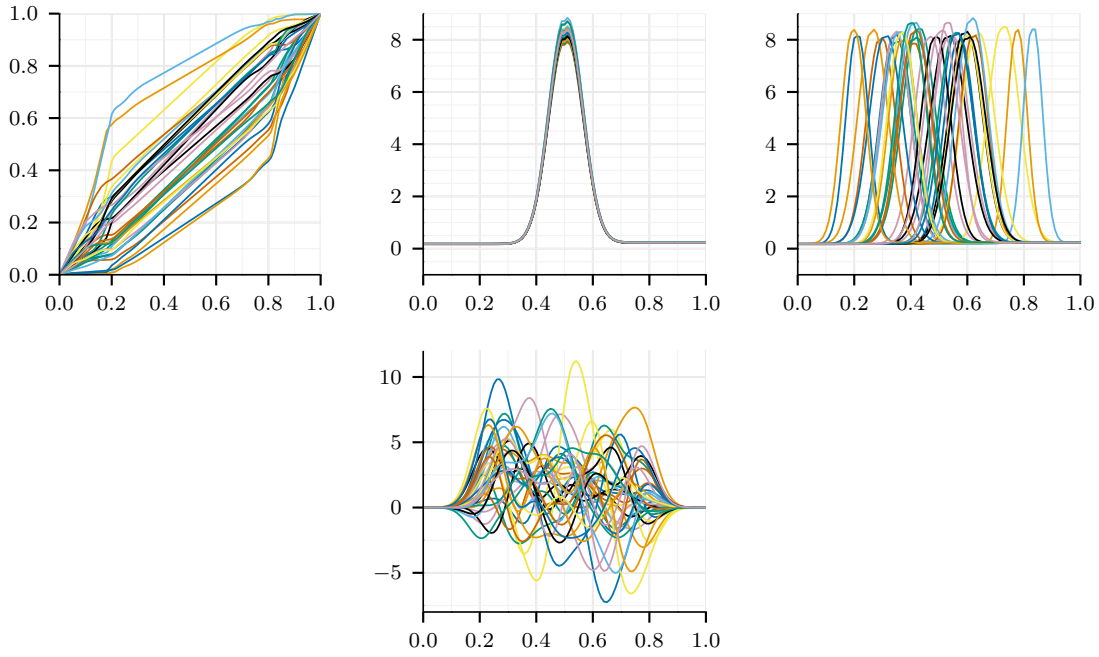


Figure 3.5: Random samples from jointly Gaussian models on fPCA coefficients of γ^s (left) and f^s (middle), and their combinations $f^s \circ \gamma^s$ (right) for Simulated Data 2. The last panel shows the random samples resulting from a Gaussian model imposed on f directly.

Amplitude-Based Classification. As described earlier, we can impose a probability model on the amplitude components data using the principal subspace associated with the aligned SRSFs. The actual model is imposed on the principal coefficients $(c_1, c_2, \ldots, c_{k_1})$, with respect to the basis $U_{h,1}, U_{h,2}, \ldots, U_{h,k_1}$. These basis elements in turn, are determined using the training data. We can select a k_1 such that the cumulative energy $\sum_{j=1}^{k_1} \Sigma_{h,jj} / \sum_{j=1}^{T+1} \Sigma_{h,jj}$ is above a certain threshold, e.g., 90 percent. There are two choices of models: Gaussian and kernel-density estimator. Classification is performed by constructing the appropriate models for each class $C_1, \cdots, C_L$ of the data. Then, for a test sample $\tilde{h}_j \in \mathbb{R}^{T+1}$ project it to the principal subspace using an orthonormal basis $U_{hl} \in \mathbb{R}^{(T+1)\times k_1}$, one for each class, and calculate the likelihood under each class. The model with the largest likelihood represents the class assigned to $\tilde{h}_j$. Therefore, our classification rule is:

$$\text{classify}(\tilde{h}_j) = \arg\max_{C_l} \; p_a(U_{hl}^{\mathsf{T}} \tilde{h}_j | K_{hl}, \mu_{hl}) \quad , \quad \text{where} \quad p_a = p_{Gauss} \quad \text{or} \quad p_{ker} \; . \tag{3.5.1}$$

Phase-Based Classification. Similarly, for the phase components, we can represent the shooting vectors, $\{v_i\}$, in a lower order dimensional space using the first k_2 columns of U_ψ. Where

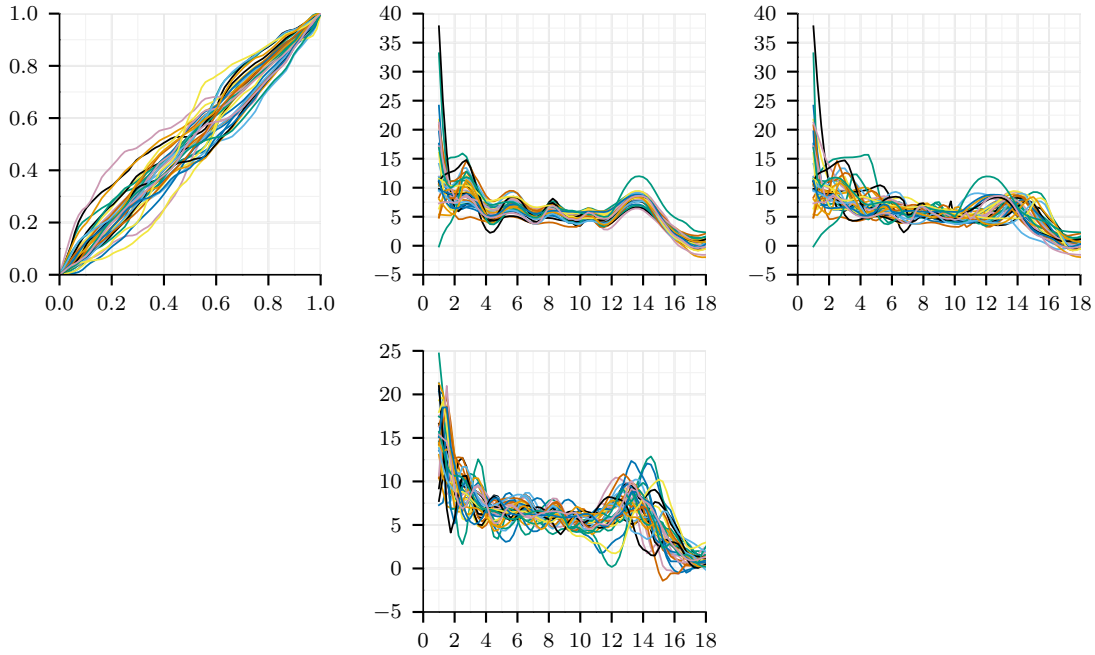


Figure 3.6: From left to right: Random samples from jointly Gaussian models on fPCA coefficients of γ^s and f^s, respectively, and their combinations $f^s \circ \gamma^s$ for the Berkley Growth data. The last panel shows the original data used in this experiment.

k_2 can be chosen similar to k_1 as described above. Once again, we can either define a Gaussian model or a kernel density estimator on these principal coefficients. We can estimate the model parameters for each class $C_1, \cdots, C_L$ using the training data. Then, for a test sample's shooting vector v_j, we project it to each model's subspace and calculate the likelihood of v_j under each pre-determined class. Therefore, our classification rule is:

$$\text{classify}(v_j) = \arg\max_{C_l} \; p_\psi(U_{\psi l}^\mathsf{T} v_j | K_{\psi l}) \quad \text{where} \quad p_\psi = p_{Gauss} \quad \text{or} \quad p_{ker} \; . \tag{3.5.2}$$

Joint Classification. Assuming independence we can combine the amplitude and phase classification rules as,

$$\text{classify}(\tilde{h}_j, v_j) = \arg\max_{C_l} \; p_a(U_{hl}^\mathsf{T} \tilde{h}_j | K_{hl}, \mu_{hl}) p_\psi(U_{\psi l}^\mathsf{T} v_j | K_{\psi l}) \tag{3.5.3}$$

and classification is as described previously.

3.6 Classification Results

In this section, we describe classifying functional data after phase and amplitude separation using pairwise distances using the distances calculated us Eq. 2.2.4 and Eq. 3.1.1 and classifying functional data using phase and amplitude models.

3.6.1 Pairwise Distances

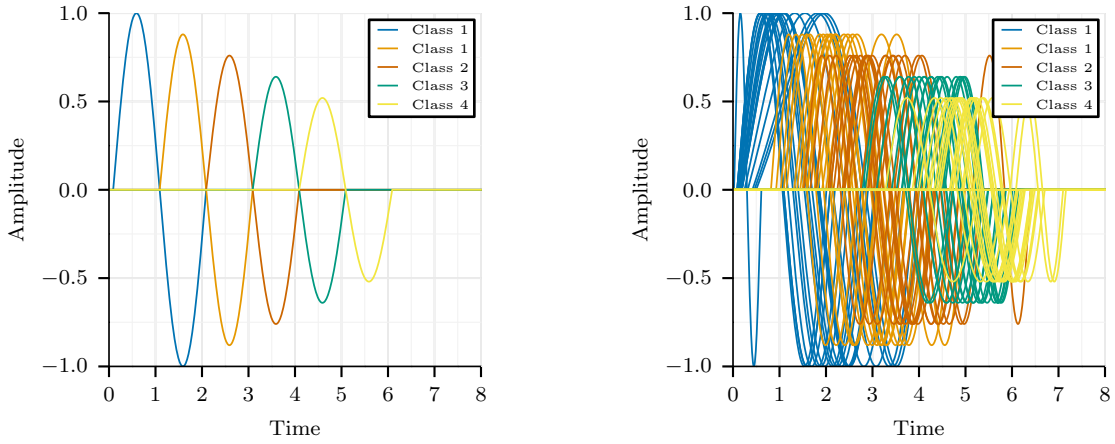
In this section, we present the classification results on a SONAR data set and a simulated data to study the effect of traditional additive noise on our metrics.

Simulated Data. We conducted a simulation study using a data set where each class is represented by a sine function and each class has a different amplitude and phase shift. We studied a five-class problem with the amplitudes for the 5 classes being 1, 0.88, 0.76, 0.64, and 0.52, respectively. The phase shift for each class was $((k-1)*\pi)$ for $k = 1, \cdots, 5$ classes and the data were generated by randomly warping the classes.

The original classes are shown in Fig. 3.7(a) and the randomly warped data for the 5 classes is shown in Fig. 3.7(b); the warping functions are generated randomly. The additive noise was generated as white Gaussian noise with mean zero and variance σ, where σ was changed for the desired signal-to-noise ratio (SNR) Then it was smoothed using a moving average with a window of size 3. The choice of the smoothing allows for numerical robustness in the calculation of the SRSFs.

The pairwise distances were calculated for the standard $\mathbb{L}^2$, d_{Naive}, d_p, and d_a and classification was performed using the leave-one-out (LOO) cross-validated nearest-neighbor classifier for varying degrees of noise. The distance d_{Naive} corresponds to the quantity $\min_\gamma \|f_1 - f_2 \circ \gamma\|$ that has often been used in the literature for functional alignment. We denote this method as a "naive" warping method and refer the associated distance matrix to as $(d_{Naive})_{ij} = ||f_i - f_j \circ \tilde{\gamma}_{ij}||$. Note that the data in the original domain does not obey the isometry property. Therefore, the distance matrix d_{Naive} is not symmetric.

Fig. 3.8 presents the classification rates for this data set versus SNR. We can see that d_a outperforms the Naive and the $\mathbb{L}^2$ methods. This implies that when the compositional noise is accounted for in the model with a proper distance, we get better classification performance. In this



(a) Original Classes

(b) Warped Data

Figure 3.7: Simulated data of 5 classes with 20 functions in each class.

example, the distance d_p expectedly gives poor performance since the same random warping was used for all the classes.

Overall, our nonlinear warping method using a proper distance performed well on the simulated data and greatly increases classification performance. Additionally, this system shows vast improvement over the standard $\mathbb{L}^2$ distance and current alignment techniques such as d_{Naive}.

SONAR Data. The data set used in these experiments were collected at the Naval Surface Warfare Center Panama City Division (NSWC PCD) test pond. For a description of the pond and measurement setup the reader is referred to [29]. The raw SONAR data was collected using a 1 - $30kHz$ LFM chirp and data was collected for nine proud targets that included a solid aluminum cylinder, an aluminum pipe, an inert $81mm$ mortar (filled with cement), a solid steel artillery shell, two machined aluminum UXOs, a machined steel UXO, a de-militarized $152mm$ TP-T round, a de-militarized $155mm$ empty projectile (without fuse or lifting eye), and a small aluminum cylinder with a notch. The aluminum cylinder is $2ft$ long with a $1ft$ diameter; while the pipe is $2ft$ long with an inner diameter of $1ft$ and 3/8 inch wall thickness.

The acoustic signals were generated from the raw SONAR data to construct target strength as a function of frequency and aspect angle. Due to the relatively small separation distances between the targets in the measurement setup, the scattered fields from the targets overlap. To

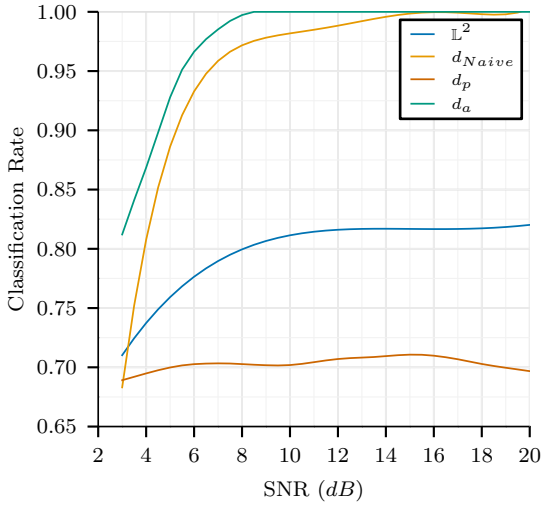


Figure 3.8: Classification rates in the presence of additive noise.

generate the acoustic templates (i.e., target strength plot of frequency versus aspect), synthetic aperture sonar (SAS) images were formed and then an inverse imaging technique was used to isolate the response of an individual target and to suppress reverberation noise. A brief summary of this process is as follows: The raw SONAR data are matched filtered and the SAS image is formed using the $\omega - k$ beamformer [59]. The target is then located in the SAS image and is windowed around selected location. This windowed image contains the information to reconstruct the frequency signals associated with a given target via inverting the $\omega - k$ beamformer [34] and the responses were aligned in rage using the known acquisition geometry. For the nine targets, 2000 different data collections runs were done, and 1102 acoustic color templates were generated using the method described above from the data set. From the acoustic color maps, one-dimensional functional data was generated by taking slices at aspect value of 0° and therefore generating 1102 data samples. We will apply our method to this SONAR data, where there are $n = 1102$ SONAR signals with nine target classes and the numbers of functions in the nine classes are $\{n_i\}_{i=1}^{9} = \{131, 144, 118, 118, 121, 119, 120, 114, 117\}$ and are sampled using 483 points. A selected subset of functions in each class from the original data is shown in Fig. 3.9. We observe that the original data are quite noisy, due to both the compositional and the additive noise, increasing variability within class and reducing separation across classes. This naturally complicates the task of target classification using SONAR signals

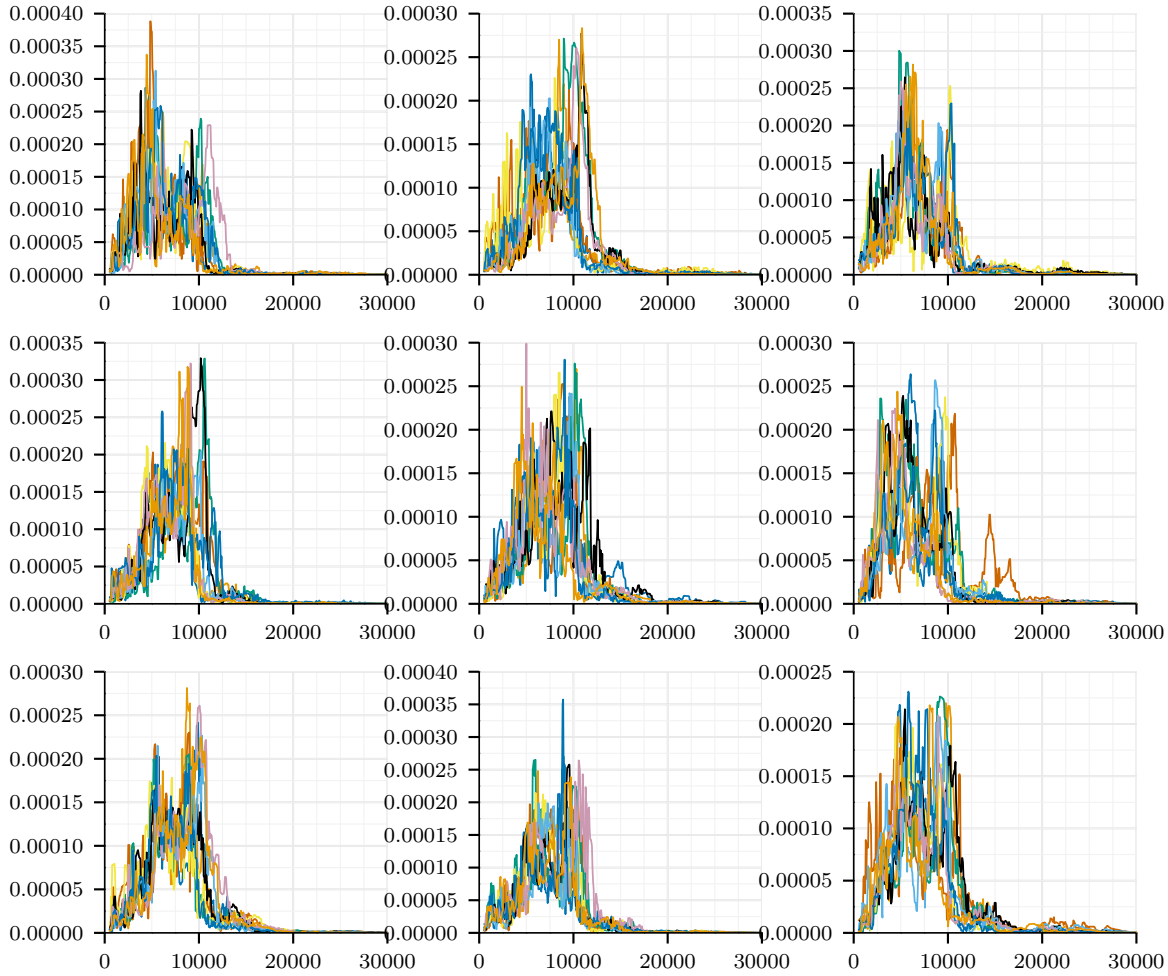


Figure 3.9: Original SONAR functions in each of the nine classes.

To have a robust estimate of the SRSF $\{q_i\}$, we first smooth the original functions 25 times $\{f_i\}$ using a standard box filter $[1/4, 1/2, 1/4]$. That is, numerically we update the signals at each discrete point by $f_i(x_k) \to \left(\frac{1}{4}f_i(x_{k-1}) + \frac{1}{2}f_i(x_k) + \frac{1}{4}f_i(x_{k+1})\right)$.

To determine the effect of smoothing on the classification performance, we conducted a small study on the number of times the smoothing filter is applied. Table 3.1 presents the classification performance versus applying the smoothing filter 0, 25, 75, 125, and 175 times. It is interesting to note that the performance is quite stable with respect to smoothing and applying the box filter 25 times gives only slightly better performance. Hence, we use that level of smoothing for each signal for the rest of the analysis.

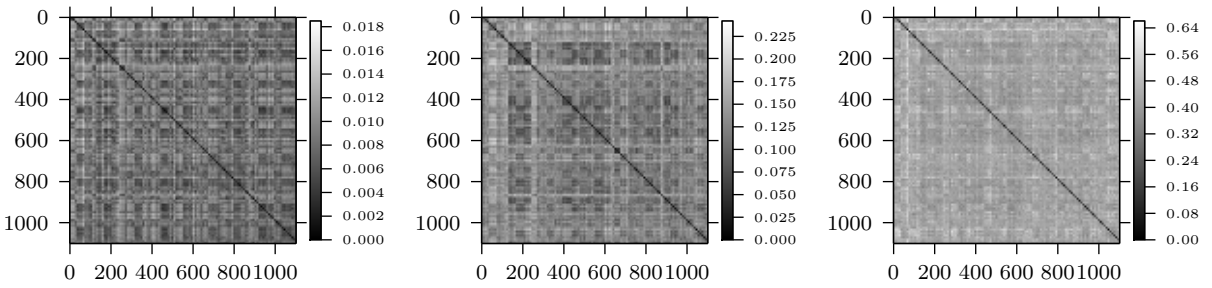
Table 3.1: Classification rates versus amount of smoothing applied.

Amount of Smoothing	0	25	75	125	175
d_p	0.57	0.58	0.59	0.58	0.55
d_a	0.63	0.73	0.67	0.64	0.60
d_{Naive}	0.61	0.64	0.62	0.57	0.51
$\mathbb{L}^2$	0.43	0.44	0.45	0.45	0.44

We first compute the standard $\mathbb{L}^2$ distance between each pair, i.e., $(\mathbb{L}^2)_{ij} = \|f_i - f_j\|, \quad i, j = 1, \ldots, n$. The matrix of pairwise $\mathbb{L}^2$ distances are shown as a gray scale image in Fig. 3.10a. This image of the pairwise distances looks very noisy, underlying the difficulty of classification using SONAR data. Based on this distance matrix, we perform classification by using the LOO cross-validation on the standard nearest-neighbor method. It is found that the accuracy is 0.44 (489/1102). We then computed distances d_a and d_p between all pairs of signals and these distance matrices are shown as gray scale images in Fig. 3.10b and 3.10c, respectively. Note that, in theory, d_p and d_a should lead to symmetric matrices but, in practice, due to the numerical errors these matrices are not exactly symmetric. So, we force them to be symmetric using $d_p \to (d_p + d_p^{\mathsf{T}})/2$, $d_a \to (d_a + d_a^{\mathsf{T}})/2$, where the superscript $^{\mathsf{T}}$ indicates the transpose of a matrix.

In the image of d_a (Fig. 3.10b), we find that the pairwise distances are more structured than the $\mathbb{L}^2$ distances. We also perform classification using the LOO cross-validated nearest-neighbor based on the d_a distances. The accuracy turns out to be 0.73 (803/1102), a significant improvement over the result (0.44) in the standard $\mathbb{L}^2$ distances. Interestingly, we find that the d_p distances also have strong indication of the target class in the data. In Fig. 3.10c, we see that the d_p image have some clusters (dark squares) along the main diagonal. The classification accuracy by d_p turns out to be 0.58 (643/1102), which is also higher than the classification performance of the standard $\mathbb{L}^2$ norm in the function space.

Since d_p and d_a each only partially describe the variability in the data, which corresponds to the phase and amplitude differences between the functions, there is a possibility of improvement if d_p and d_a are used jointly. One simple idea is to linearly combine these two distances and use the weighted distance to perform classification on the data. Here the amplitude and phase are being treated as two different "features" of the signals. To accurately represent the contribution from each distance, we first normalize d_p and d_a by the maximum values in the matrices, respectively.



(a) $\mathbb{L}^2$ (b) d_a (c) d_p

Figure 3.10: The pairwise distances using the $\mathbb{L}^2$ (a), d_a (b), and d_p (c) metrics.

That is, $d_p \to \frac{d_p}{\max d_p}$, $d_a \to \frac{d_a}{\max d_a}$. Then, for $\tau \in [0, 1]$, we define

$$d_\tau = \tau d_p + (1 - \tau) d_a.$$

d_τ is a weighted average of d_p and d_a with $d_0 = d_a$ and $d_1 = d_p$.

The next step is the estimation of an optimal τ. Towards this end, we randomly select 50% of the given signals as training data and evaluate the LOO classification performance for different values of τ. Since this selection is random, the resulting evolution is potentially random. Figure 3.11a shows the performance profile versus τ for 100 randomly selected training data. An average of these curves is superimposed on the same plot (thick solid line). A histogram of the optimal values of τ for different random selections of the training data is shown in b. Both these figures show that a broad range of τ values, from 0.3 to 0.7, all result in a decent increase in the classification performance over the individual metrics d_p and d_a, and the general pattern of increase is similar. In fact, if we use the full data and plot the LOO classification performance versus τ, we obtain the plot shown in c. The overall shape (and the location of the maximizer) of this curve is very similar to the curves in a and underscores the independence of different observations. From this study, we select a value, say $\tau = 0.41$ and use that to perform LOO classification on the full data.

When $\tau = 0.41$ is used, we get an accuracy of 0.76 (839/1102), which is higher then the accuracy of the $\mathbb{L}^2, d_a$, and d_p distances. This indicates that the variability in the SONAR signals are better characterized when we separate the phase and amplitude variabilities, and better classification can be achieved when both variabilities are utilized.

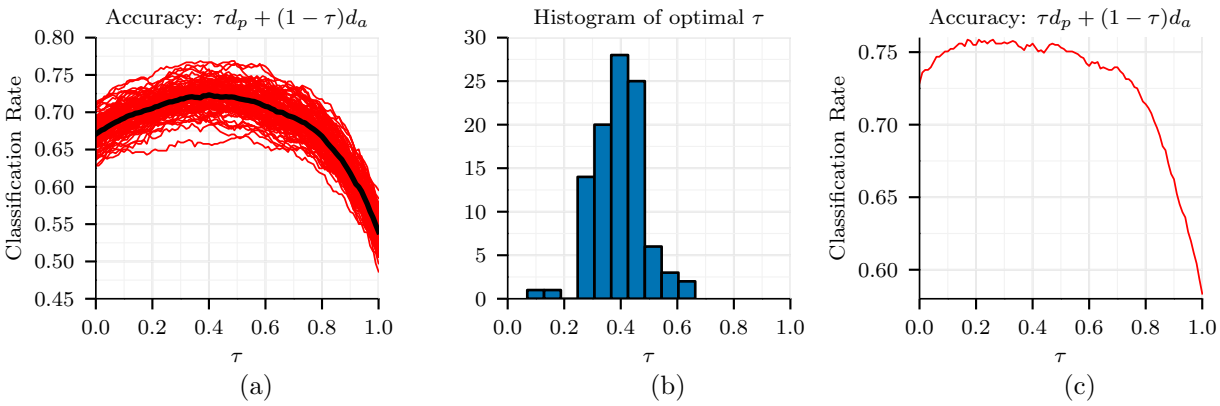


Figure 3.11: (a) Evolution of classification performance versus τ for randomly selected training data. The average of these curves is drawn on the top. (b) The histogram of optimal τ values for different random selections of training data. (c) Overall performance versus τ for the full data.

Next, we compute the "naive" distance between any two signals presented in the previous section, according to $(d_{Naive})_{ij} = \arg\min_{\gamma\in\Gamma} \|f_i - f_j \circ \gamma\|$. We also perform the cross-validated nearest-neighbor using the d_{Naive} and find that the accuracy is 0.64 (702/1102). This is slightly better than the accuracy by d_p, but worse than that of d_a. This indicates that even a simple method of warping can help remove certain warping noise in the SONAR data. However, the performance is not as good as the more formal SRSF-based warping.

Next, we generated a cumulative match characteristic (CMC) curve [5] for the distances d_p, d_a, $d_\tau(\tau = 0.41)$, d_{Naive}, and $\mathbb{L}^2$. A CMC curve plots the probability of classification against the returned candidate list size and is presented in Fig. 3.12. Initially, d_a and d_τ outperform the other distances with d_{Naive} slightly outperforming d_p. After a slight increase in the returned list size, d_p begins to outperform d_{Naive} and our method rapidly approaches over 0.90 classification rate, in contrast to the d_{Naive} and the standard $\mathbb{L}^2$ distances.

3.6.2 Phase and Amplitude Models

In this section, we present the classification results on a signature data [75], an iPhone-generated action data set from [44], and the SONAR data set using models developed using vertical and horizontal fPCA.

Signature Data. In this section, we test our classification method on a signature recognition data set from [75]. The data was captured using a WACOM Intuos tablet. The data set consists

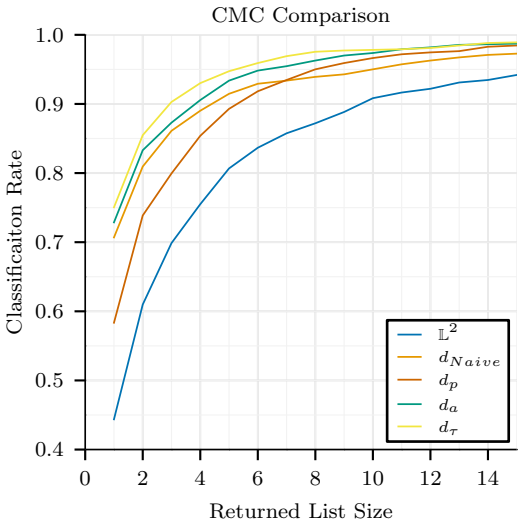


Figure 3.12: CMC Comparison of $\mathbb{L}^2$, d_{Naive}, d_p, d_a and the weighted d_τ ($\tau = 0.41$) distances for 0° aspect angle.

of signature samples from 40 different subjects with 20 real signature samples of the subject and another 20 samples, which are forgeries of the subject's signature. In our analysis we are going to distinguish between the real and forged signature for two of the subjects using the tangential acceleration. The tangential acceleration is computed as $f_i(t) = \sqrt{[X_i''(t)]^2 + [Y_i''(t)]^2}$. To again have a robust estimate of the SRSF $\{q_i\}$, we first smooth the original signals 100 times $\{f_i\}$ using the standard box filter described previously. The smoothed acceleration functions are aligned in each class (real vs. fake) using the alignment algorithm from Chapter 2.2.3 (Algorithm 2.1). An example signature with 10 realizations is shown in Fig. 3.13 along with the corresponding acceleration functions for both the real and fake signatures, the corresponding aligned functions, and warping functions.

Models were generated for the three classes, as was outlined in Section 3.3, by performing vertical and horizontal fPCA on the aligned data and the warping functions, respectively. We then impose a multivariate Gaussian model, p_{Gauss}, on the reduced data for each class, it is assumed here that the cross-covariances L_1 and L_2 are zero. The threshold to select the number of dimensions, k_1 and k_2, was set at 95%. Classification for just the amplitude component was performed as described in Section 3.6 using the classification rule in (Eqn. 3.5.1) and was evaluated using 5-fold cross-validation. Similarly, the classification rule in (Eqn. 3.5.2) was used for the phase compo-

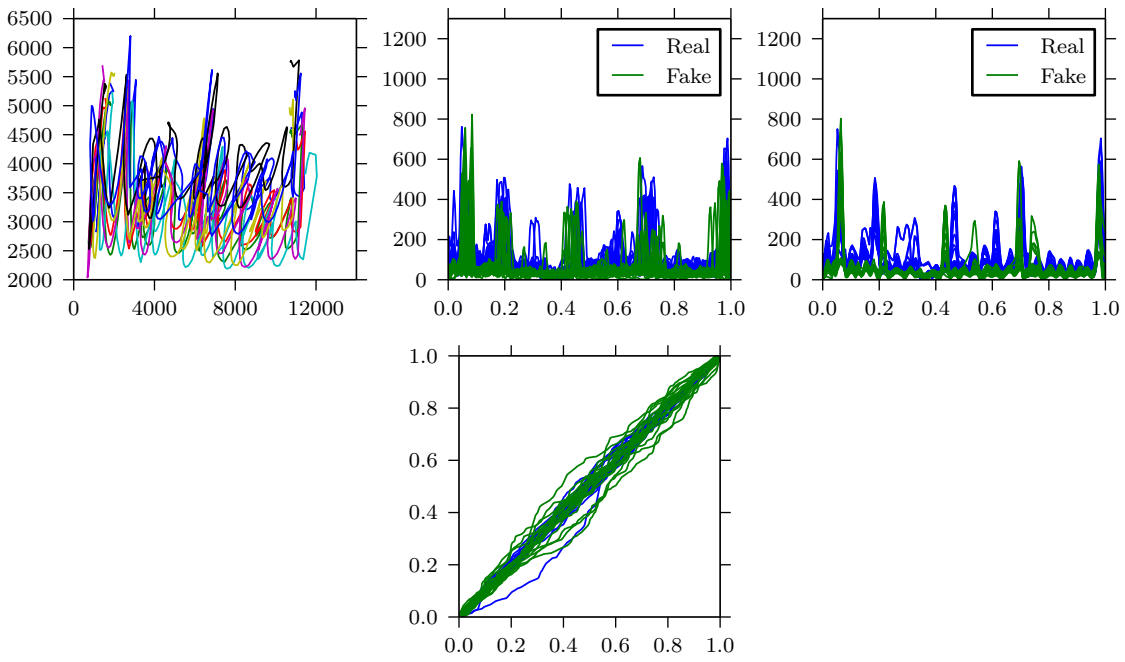


Figure 3.13: From left to right: the original signature samples for one of the subjects, the corresponding tangential acceleration functions for both the real and fake signatures, the corresponding aligned functions, and warping functions.

nent. Moreover, the joint classification was performed using (Eqn. 3.5.3). Table 3.2a presents the the mean and standard deviation (shown in parentheses) of the classification rates from the cross-validation for the three rules. It also compares to the standard $\mathbb{L}^2$ where models were generated directly on the original data, dimension reduction with fPCA, and imposing a multivariate normal distribution. Corresponding results for another subject, U13 is presented in Table 3.2b.

The classification rates have a low standard deviation indicating good generalization, though there is a little variation for the phase only model. For both subjects the amplitude only rule greatly outperforms both the phase only rule and the standard $\mathbb{L}^2$ with the best performance of 93% and 75% for subjects U1 and U13, respectively. As the phase only rule performs poorly combining it with the amplitude only rule it brings down the overall performance. The alignment and modeling using a proper distance improves the overall classification performance of the data. To compare the results between p_{Gauss} and p_{kern}, we classified the data again forming models using p_{kern} which was discussed in Section 3.6, where each of the k_1 and k_2 components has an estimated density using a kernel density estimator and independence is assumed. We used the Gaussian kernel function

Table 3.2: Mean classification rate and standard deviation (in parentheses) for 5-fold cross-validation on the signature data.

(a) Subject U1

	Gaussian	Kernel Density
amplitude only	0.93 (0.07)	0.78 (0.19)
phase only	0.65 (0.16)	0.75 (0.09)
phase and amplitude	0.90 (0.05)	0.80 (0.07)
standard $\mathbb{L}^2$	0.60 (0.14)	0.55 (0.11)

(b) Subject U13

	Gaussian	Kernel Density
amplitude only	0.75 (0.14)	0.78 (0.21)
phase only	0.50 (0.01)	0.50 (0.01)
phase and amplitude	0.58 (0.11)	0.60 (0.10)
standard $\mathbb{L}^2$	0.50 (0.01)	0.53 (0.06)

and the bandwidth was selected automatically based upon the data using the method presented by [6]. Classification using the three classification rules were performed using 5-fold cross- validation. Table 3.2a and Table 3.2b present the the mean and standard deviation of the classification rates from the cross-validation for the three rules as well as comparing to the standard $\mathbb{L}^2$. Models were generated directly on the original data using fPCA and the kernel density estimator for subjects U1 and U13, respectively. We see an improvement in the phase only method for subject U1 and reduction in performance for the other methods; this suggest the warping functions have some non-Gaussian behavior. However, for subject U13, there is a minimal change between the Gaussian and kernel estimator.

iPhone Action Data. This data set consists of aerobic actions recorded from subjects using the Inertial Measurement Unit (IMU) on an Apple iPhone 4 smartphone. The IMU includes a 3D accelerometer, gyroscope, and magnetometer. Each sample was taken at $60Hz$, and manually trimmed to 500 samples ($8.33s$) to eliminate starting and stopping movements and the iPhone is always clipped to the belt on the right hand side of the subject. There is a total of 338 functions for each measurement on the IMU and the actions recorded consisted of biking, climbing, gym bike,

jumping, running, standing, treadmill, and walking. The number of samples being 30, 45, 39, 45, 45, 45, 44, and 45, respectively for each action. For more information on the data set the reader is referred to [44]. For our experiments we used the accelerometer data in the x-direction. Again, to have a robust estimate of the SRSF $\{q_i\}$, we first smooth the original signals 100 times $\{f_i\}$ using the standard box filter described in the previous section. As with the previous data set, the smoothed iPhone data is aligned in each class (activity) using our method. A selected subset of functions from three activities is shown in Fig. 3.14 along with corresponding aligned functions and warping functions.

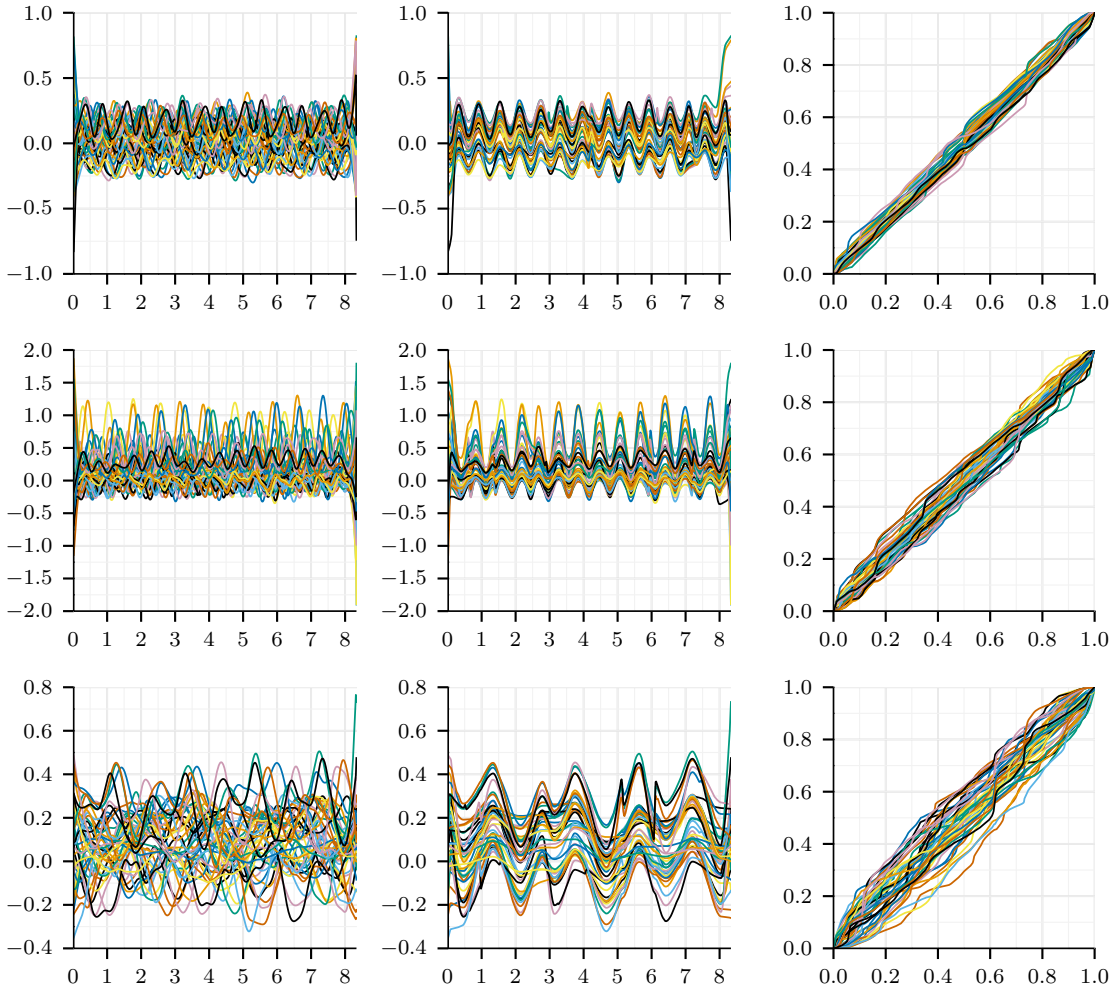


Figure 3.14: Original iPhone functions for the walking, jumping, and climbing activities in the first column (in corresponding descending order) with the corresponding aligned functions and warping functions in the second and third columns, respectively.

To perform the classification, models were generated for the 8 classes by performing vertical and horizontal fPCA on the aligned data and the warping functions; then imposing a multivariate Gaussian on the reduced data for each class. The threshold to select the number of dimensions, k_1 and k_2, was set at 95%. Classification was performed as in the previous section. Table 3.3 presents the mean and standard deviation of the classification rates for the cross-validation for all three rules as well as comparing to the standard $\mathbb{L}^2$. The classification rates have a low standard deviation indicating good generalization. The phase only rule and the amplitude only rule, drastically out-perform the standard $\mathbb{L}^2$ with the combination providing the best performance at 62%. The alignment and modeling using a proper distance improves the overall classification performance of the data. We again used the kernel density estimator to compare the results with the Gaussian kernel and the results are presented in Table 3.3. Using the kernel density estimator we see only minor improvements in the phase only rule, suggesting the Gaussian assumption is sufficient for this data.

Table 3.3: Mean classification rate and standard deviation (in parentheses) for 5-fold cross-validation on the iPhone data.

	Gaussian	Kernel Density
amplitude only	0.60 (0.04)	0.62 (0.05)
phase only	0.34 (0.06)	0.35 (0.06)
phase and amplitude	0.62 (0.08)	0.62 (0.07)
standard $\mathbb{L}^2$	0.12 (0.02)	0.12 (0.02)

SONAR Data. The data set used in these experiments was the sonar data as described in Section 3.6.1. As with the previous data sets, the smoothed SONAR data is aligned in each class using our method. Models were generated for the three classes by performing vertical and horizontal fPCA on the aligned data and the warping functions then, imposing a multivariate Gaussian on the reduced data for each class, with the aligned data shown in Fig. 3.15. The threshold to select the number of dimensions, k_1 and k_2, was set at 90%. Table 3.4 presents the classification rates for the cross-validation for all three rules as well as comparing to the standard $\mathbb{L}^2$.

The classification rates have low standard deviation indicating good generalization for the SONAR data. The phase only rule and the amplitude only rule out perform the standard $\mathbb{L}^2$ with the combination providing the best performance at 54%. The alignment and modeling using

Table 3.4: Mean classification rate and standard deviation (in parentheses) for 5-fold cross-validation on SONAR data.

	Gaussian	Kernel Density
amplitude only	0.44 (0.03)	0.47 (0.02)
phase only	0.42 (0.02)	0.43 (0.02)
phase and amplitude	0.54 (0.03)	0.53 (0.03)
standard $\mathbb{L}^2$	0.33 (0.01)	0.34 (0.02)

a proper distance improves the overall classification performance of the data. We again used the kernel density estimator to compare the results with the Gaussian assumption and the results are presented in Table 3.4. Using the kernel density estimator we see improvements in the classification results. However, there is not a dramatic improvement suggesting the Gaussian assumption is sufficient for this data.

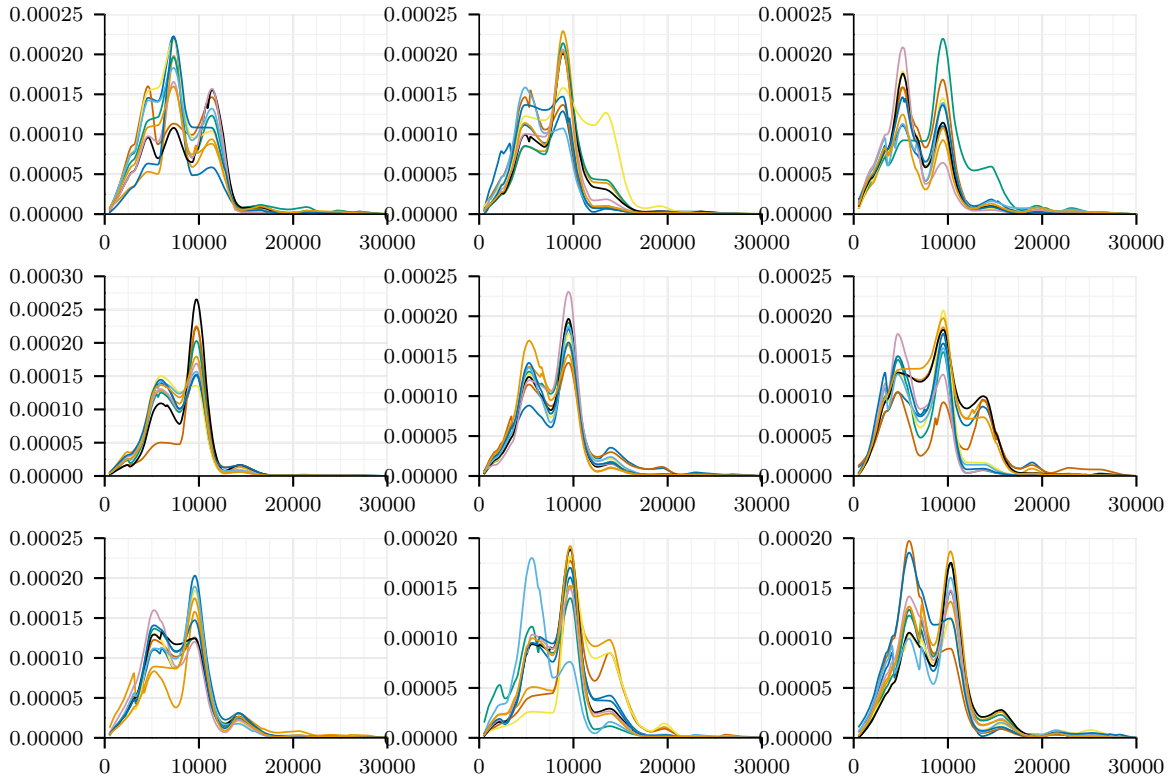


Figure 3.15: Aligned and smoothed SONAR functions in each of the nine classes.

CHAPTER 4

JOINT ALIGNMENT AND COMPONENT ANALYSIS

This chapter discusses the problem of *elastic* component analysis, in which we incorporate the phase-variability into the objective function of the component analysis. This, in turn, alters how the the optimization is performed. The warping functions are computed at the same time the corresponding components are extracted. The solution is a more natural approach to the analysis as the alignment is not a "pre-processing" step, but rather part of the complete solution using one metric.

We assume that we have the functions and their corresponding square-root slope functions (SRSFs) as described in the Chapter 2. We have two goals in this chapter: First, is to perform elastic functional principal component analysis (fPCA) where the objective is to maximize the variance in data that has phase-variability. Second, is to perform elastic functional partial least squares (fPLS) where the objective is to maximize the covariance between two functions, f and g, which both contain phase-variability.

4.1 Functional Principal Component Analysis

The motivation for fPCA is that the directions of high variance will contain more information then direction of low variance. Let $f_1, \cdots, f_n$ be a given set of functions, the optimization problem for fPCA can be written as

$$\min_{w_i} E\|f - \hat{f}\|^2 \tag{4.1.1}$$

where $\hat{f} = \mu_f + \sum_{i=1}^{n} \beta_i w_i(t)$ is the fPCA approximation of f with corresponding mean μ_f, $\beta_i = \int (f - \mu_f) w_i(t)\, dt$, and basis functions $\{w_i(t)\}$. The expectation is taken over f.

Since our desire is to obtain a warping invariant fPCA we now introduce warping of the functions into Eq. 4.1.1 as

$$\min_{\{w_i\}} E\left[\min_{\gamma} \|f \circ \gamma - (\mu_f + \sum_{i=1}^{n} \alpha_i w_i)\|^2\right] \tag{4.1.2}$$

where $\alpha_i = \int (f_i \circ \gamma - \mu_f) w_i(t)\, dt$. The problem with Eq. 4.1.2 is that cost function is neither symmetric nor positive definite. Moreover if we look at the sample version of Eqn. 4.1.2

$$\min_{\{w_i\}} \frac{1}{N} \sum_{j=1}^{N} \left[\min_{\gamma_j} \| f_j \circ \gamma_j - (\mu_f + \sum_{i=1}^{n} \alpha_{ji} w_i(t)) \|^2 \right],$$

we have the pinching problem as demonstrated in Chapter 2. For example, if the functions f_i have a point in their ranges that is common, an optimal set $\{\gamma_i^*\}$ can be found such that $\mu_f(t)$ will become a constant and $\hat{f}$ will be zero. In this case the cost function will become zero and the solution is degenerate. An simple example of this problem is shown in Fig. 4.1 where the original functions are a bounded half-period sine function, $\sin(2\pi t)$, randomly shifted along the x-axis. The original functions, aligned functions $\{f_i \circ \gamma_i\}$, and optimal warping functions $\{\gamma_i^*\}$ are shown in Panels a, b, and c, respectively. In this case the warping functions spread the functions out such that they are identical and the mean becomes a constant. This solution is degenerate as the principal components are zero and are not representative of the original data.

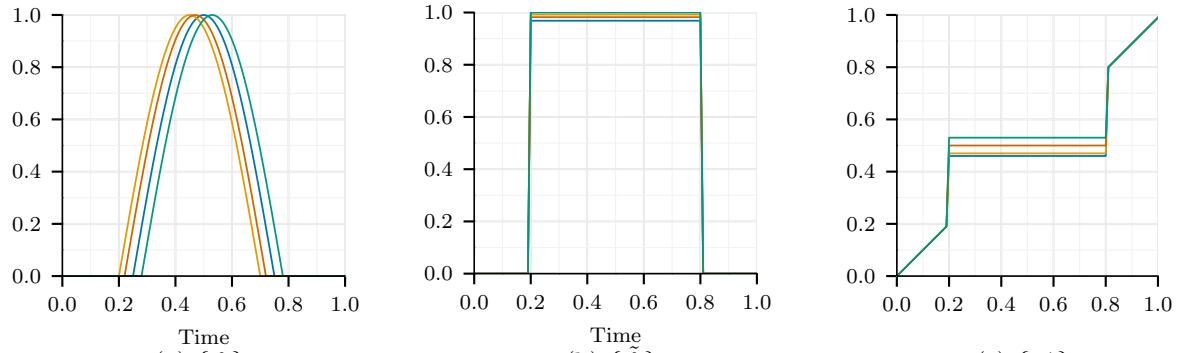


Figure 4.1: Example of pinching problem in functional principal component analysis with warping.

To address this problem, we use the extended Fisher-Rao metric and the *square-root slope function* or SRSF of f, where the Fisher-Rao metric is the standard $\mathbb{L}^2$ metric, as was demonstrated in Chapter 2.2.2. Since the metric is isometric, it is invariant to simultaneous warping of the inputs and avoids the above mentioned problems

$$\min_{\{w_{q_i}\}} E \left[\min_{\gamma} \| (q, \gamma) - (\mu_q + \sum_{i=1}^{n} \alpha_i w_{q_i}) \|^2 \right] \tag{4.1.3}$$

where $\alpha_i = \int((q_i, \gamma_i) - \mu_q) w_{q_i}(t)\, dt$. This is different from doing fPCA on functions after alignment as was done previously in Chapter 3, since we are simultaneously optimizing over the principal coordinates and the warping.

In practice, where we have an ensemble set functions, the expectation is replaced with a summation

$$H_{pca} = \min_{\{w_{q_i}\}} \frac{1}{N} \sum_{j=1}^{N} \left[\min_{\gamma_j} \|(q_j, \gamma_j) - (\mu_q + \sum_{i=1}^{n} \alpha_{ji} w_{q_i})\|^2 \right]. \tag{4.1.4}$$

The algorithm for computing this minimization is given in Algorithm 4.1, where it is done by a two part process where the fPCA is calculated for the current set of SRSFs and then the SRSFs are aligned to the principal components. In this process the fPCA is computed using the vertical fPCA as explained in Chapter 3.2. Note that if $\hat{q}$ is a minimizer of the cost function, then so is $\hat{q} \circ \gamma$ for any $\gamma \in \Gamma$ since the cost function is invariant to random warpings of its input variables. So, we have an extra degree of freedom in choosing an arbitrary element of the set $\{\hat{q} \circ \gamma | \gamma \in \Gamma\}$. To make this choice unique, we can define a special element of this class as follows. Let $\{\gamma_i^*\}$ denote the set of optimal warping functions, one for each i, in Eqn. 4.1.4. Then, we can choose the $\hat{q}$ to that element of its class such that the mean of $\{\gamma_i^*\}$, denoted by γ_μ, is γ_{id}, the identity element. (The notion of the mean of warping functions and its computation are described in Algorithm 3.1.)

This procedure results in five items:

1. μ_f, the mean function,
2. $\{\tilde{q}_i\}$, the set of aligned SRSFs,
3. $\{\gamma_i^*\}$, the set of optimal warping functions,
4. $\{\tilde{f}_i\}$, the set of aligned functions, and
5. $\{w_{f_i}(t)\}$, the principal eigenfunctions

One can then use this framework to visualize the vertical principal-geodesic paths. The basic idea is to compute a few points along geodesic path $\tau \mapsto \mu + \tau\sqrt{\Sigma_{jj}} w_j$ for $\tau \in \mathbb{R}$ in $\mathbb{L}^2$, where Σ_{jj} and w_j are the j^{th} singular value and eigenfunction, respectively.

A test of the convergence of the algorithm was done on the two peak simulated data set from Fig. 2.5. The algorithm converged in 6 iterations and the cost function is shown in Fig. 4.2.

Algorithm 4.1 Simultaneous Alignment and Extraction of Principal Components

Initialization Step: Select $\mu_q = q_i$, where $i = \arg\min_{1 \leq i \leq n} ||q_i - \frac{1}{n}\sum_{j=1}^{n} q_j||$, set $\{\gamma_i\} = \gamma_{id}$, set $l = 1$.

while $\|H^{(l+1)} - H^{(l)}\|^2 < \epsilon$ **do**

  Compute $K_h = \frac{1}{n-1}\sum_{i=1}^{n} E[(\tilde{h}_i - \mu_h)(\tilde{h}_i - \mu_h)^\mathsf{T}]$

  Take the SVD of $K_h = U_h \Sigma_h V_h^\mathsf{T}$ and set w_i, $i = 1, \ldots, n$ be the n left singular vectors

  Compute $\alpha_{ji} = \int ((q_j, \gamma) - \mu_q) w_{q_i}(t)\, dt \ \forall j$

  For each q_i find the $\gamma_i^{(l)*}$ such that

$$\gamma_i^{(l)*} = \arg\min_{\gamma \in \Gamma} \left(\|(q_i, \gamma_i) - (\mu_q + \sum_{j=1}^{n} \alpha_{ij} w_{q_j})\|^2 \right).$$

  The solution to this optimization comes from the dynamic programming algorithm.

  Compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^{(l)*})\sqrt{\dot{\gamma}_i^{(l)*}}$.

  Update the mean using $\mu_q \mapsto \frac{1}{n}\sum_{i=1}^{n} \tilde{q}_i$

end while

Update $\gamma_i^* \mapsto \gamma_i^{(1)*} \circ \gamma_i^{(2)*} \circ \cdots \circ \gamma_i^{(l)*}$

The function μ represents a whole equivalence class of solutions and now we select the preferred element μ_q of that orbit:

1. Compute the mean γ_μ of all $\{\gamma_i^*\}$ (using Algorithm 3.1) Then compute $\mu_q = (\mu \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma}_\mu^{-1}}$
2. Update $\gamma_i^* \mapsto \gamma_i^* \circ \gamma_\mu^{-1}$
3. Update $w_{q_j} = (w_{q_j} \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma}_\mu^{-1}}$ Then compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$

Map the SRSFs, μ_q, and w_q back to the function space $\mathcal{F}$ using $\tilde{f}_i(t) = f_i(t_0) + \int_{t_0}^{t} \tilde{q}_i(s)|\tilde{q}_i(s)|\, ds$

4.1.1 Numerical Results

To illustrate the developed fPCA method, we run the algorithm on the data used previously in [37] and as was explained in Chapter 2.2.2. As a reminder, the individual functions are given by: $y_i(t) = z_{i,1}e^{-(t-1.5)^2/2} + z_{i,2}e^{-(t+1.5)^2/2}$, $t \in [-3, 3]$, $i = 1, 2, \ldots, 21$, where $z_{i,1}$ and $z_{i,2}$ are *i.i.d.* normal with mean one and standard deviation 0.25. (Note that although this framework was developed for functions on $[0, 1]$, it can easily be adapted to an arbitrary interval). Each of these functions is then warped according to: $\gamma_i(t) = 6(\frac{e^{a_i(t+3)/6}-1}{e^{a_i}-1}) - 3$ if $a_i \neq 0$, otherwise $\gamma_i = \gamma_{id}$ ($\gamma_{id}(t) = t$ is the identity warping). Here a_i are equally spaced between -1 and 1, and the observed functions are computed using $f_i(t) = y_i(\gamma_i(t))$.

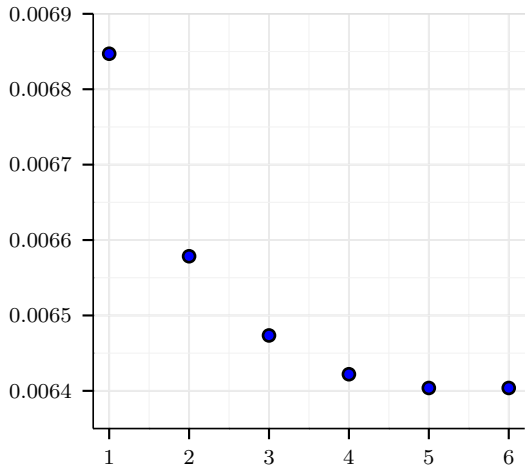
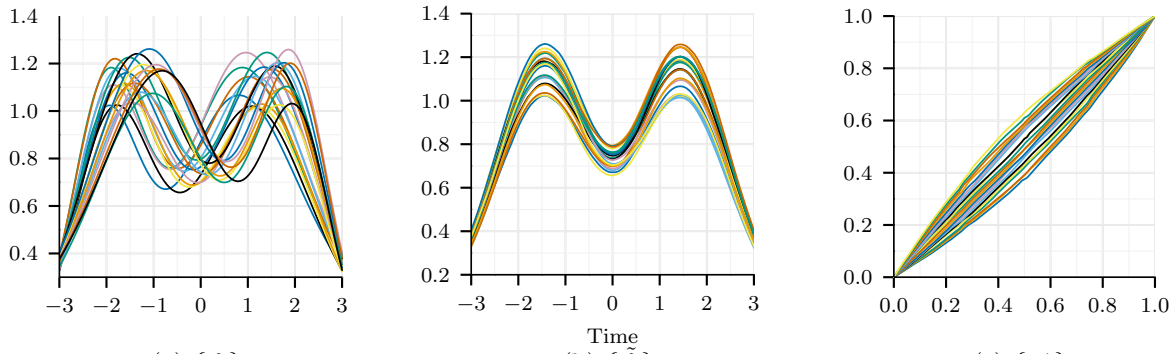


Figure 4.2: Evolution of cost function for Algorithm 4.1.

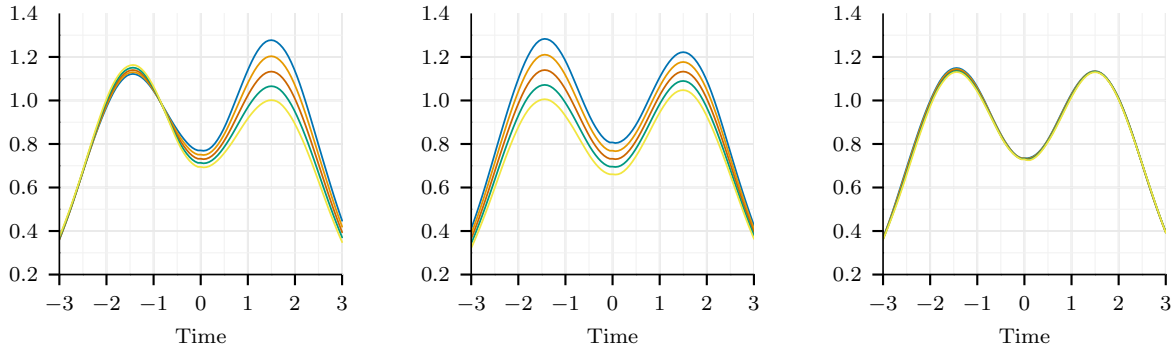
A set of 21 such functions forms the original data and is shown in Panel a of Fig. 4.3. Panel b presents the resulting aligned functions using our method in Algorithm 4.1 $\{\tilde{f}_i\}$ and Panel c plots the corresponding $\{\gamma_i^*\}$ using three principal components. It is apparent that the plot of $\{\tilde{f}_i\}$ shows a tighter alignment of functions with sharper peaks and valleys and thinner bands around the mean. This indicates that the effects of warping generated by the γ_is have been completely removed and only the randomness from the y_is remains. Fig. 4.4 presents the principal-geodesic paths, $f_{\tau,j}$ for $\tau = -2, -1, 0, 1, 2$ and $j = 1, 2, 3$, in Panels a, b, and c, respectively. The first principal direction mainly corresponds to the height variation of the second peak, while the second principal component mostly captures the height variation of the first peak. The later components have relatively negligible variance.

We then evaluated the algorithm on the Berkley growth data, specifically the male growth velocity curves. A set of 39 such functions form the original data and is shown in Panel a of Fig. 4.5. Panel b presents the resulting aligned functions using our method in Algorithm 4.1 $\{\tilde{f}_i\}$ and Panel c plots the corresponding $\{\gamma_i^*\}$ using three principal components. Again, we see an overall good alignment of the data. Fig. 4.6 presents the principal-geodesic paths, $f_{\tau,j}$ for $\tau = -2, -1, 0, 1, 2$ and $j = 1, 2, 3$, in Panels a, b, and c, respectively. The first principal direction mainly corresponds to the initial value of the function, the second component contains peak variation, and the third component contains overall variation.



(a) $\{f_i\}$ (b) $\{\tilde{f}_i\}$ (c) $\{\gamma_i^*\}$

Figure 4.3: Alignment results on simulated data from Algorithm 4.1.

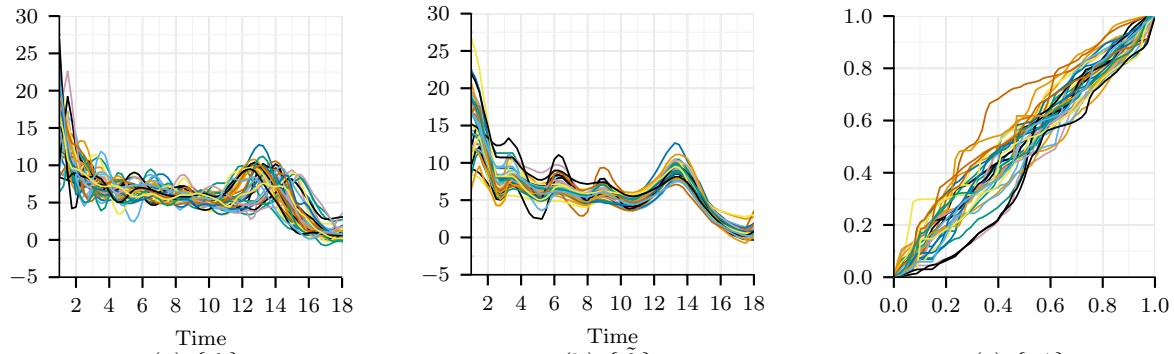


(a) PD 1 (b) PD 2 (c) PD 3

Figure 4.4: Principal directions on simulated data from Algorithm 4.1.

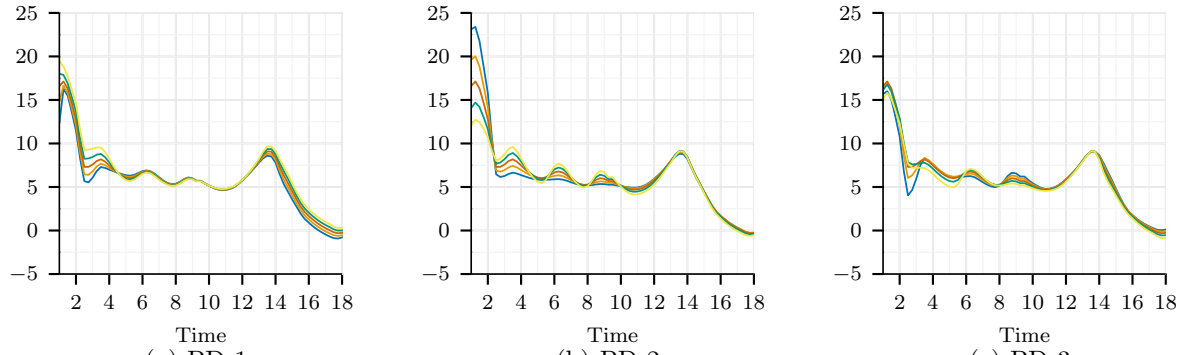
Next, we compare our results to a similar method described by Kneip and Ramsay [37] for both the simulated and growth data. The difference between their method and ours is the SRSF representation and the use of a proper distance. In [37] they perform the analysis in $\mathcal{F}$ space using the standard $\mathbb{L}^2$ metric, which is not a proper distance. Fig. 4.7 presents alignment results using the method described in [37] with the corresponding principal directions in Fig. 4.8. The alignment is good and comparable to our method. However, the principal directions that are provided are different from those from our method in Fig. 4.4. Both the first and second principal directions contain the variability in both peaks.

Next, we evaluated the growth data and Fig. 4.9 presents alignment results using Kneip and Ramsay's method with the corresponding principal directions in Fig. 4.10 using the same number



(a) $\{f_i\}$ (b) $\{\tilde{f}_i\}$ (c) $\{\gamma_i^*\}$

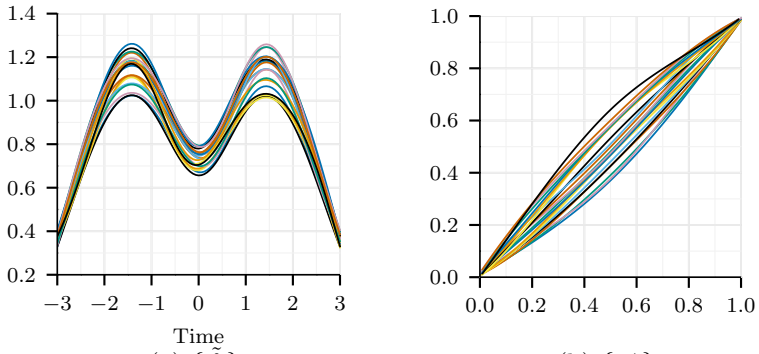
Figure 4.5: Alignment results on Berkley Growth data from Algorithm 4.1.



(a) PD 1 (b) PD 2 (c) PD 3

Figure 4.6: Principal directions on Berkley Growth data from Algorithm 4.1.

of principal components (3) as our method. First, visually the removal of the phase-variability is not as good as our method, since a few of the peaks are not aligned. The principal directions are vastly different from those produced using our method in Fig. 4.6, with all three principal directions having variability across the functions. The differences in the results are largely due to two reasons: 1) Kneip and Ramsay use the MSE alignment method in their algorithm which has initialization issues and lacks performance using more complicated data sets as was shown in Chapter 2.2.3. 2) They use the standard $\mathbb{L}^2$ metric, which is not isometric and is not the right type of metric to perform this analysis. Therefore, using the SRSF framework and a proper metric shows increased performance over current published methods.



(a) $\{\tilde{f}_i\}$ (b) $\{\gamma_i^*\}$

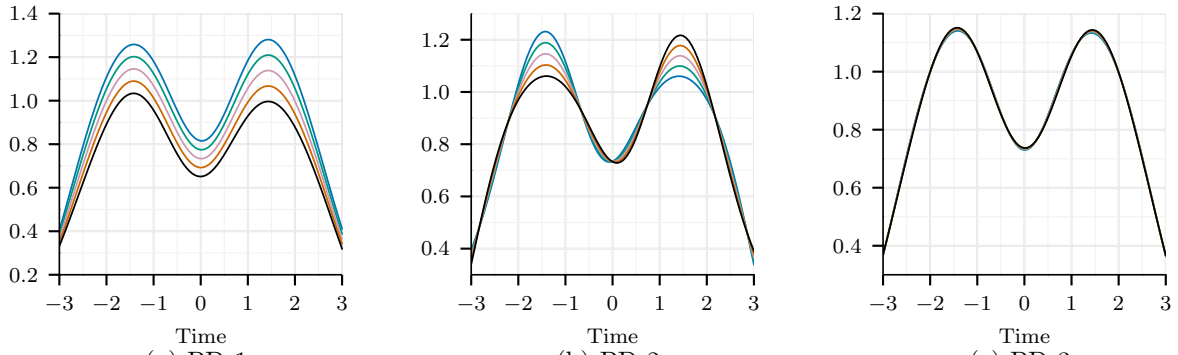
Figure 4.7: Alignment results on simulated data using Kneip and Ramsay's method described in [37].

We can also quantify the alignment performance using the decrease in the cumulative cross-sectional variance of the aligned functions. Using the metrics from Chapter 3 we can compare the phase- and amplitude-variances for the elastic method and Kneip and Ramsay's, which are listed below in Table 4.1 with the simulated unimodal data on the top row and the growth data on the bottom row:

Table 4.1: The comparison of the amplitude variance and phase variance for different fPCA algorithms on the Simulated and Berkley Growth data set.

Data	Component	Original Variance	Elastic Method	Kneip
Simulated	Amplitude-variance	0.083	**0.018**	0.019
	Phase-variance	0	0.062	0.060
Berkley Growth	Amplitude-variance	42.32	**20.06**	24.65
	Phase-variance	0	41.29	27.25

The alignment performance for the simulated data is comparable between the two methods as both methods achieve nearly the same amplitude- and phase-variance. However, for the more difficult Berkley growth curves our method out performs the other method. This can be seen by the larger increase in the phase-variance which attributes to a better alignment of the data. The amplitude variance is also greatly reduced which also indicates a tighter alignment of functions.

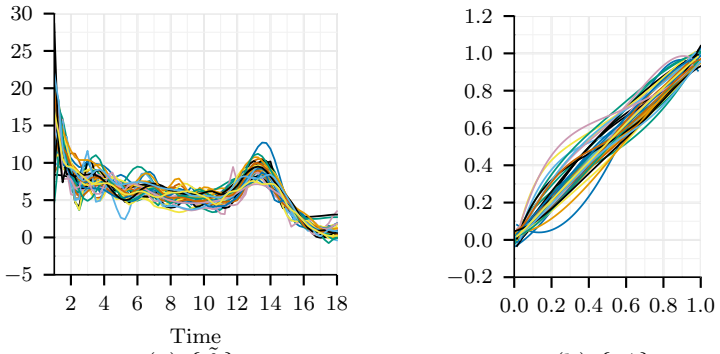


(a) PD 1 (b) PD 2 (c) PD 3

Figure 4.8: Principal directions on simulated data using using Kneip and Ramsay's method described in [37].

Table 4.2 presents the resulting singular values as a percentage of the cumulative energy resulting form the elastic method in Algorithm 4.1 and Kneip and Ramsay's method from [37]. Panel a presents the singular values resulting from the simulated data and we see that Kneip and Ramsay's method packs all of the energy into the first component and the elastic method is in the first 4 components. All of the energy clearly should not be attributed to the first component as there is more than one type of variation in the data. Panel b presents the singular values resulting from the Berkley Growth data and we see that Kneip and Ramsay's method only obtains 70% of the energy in the first four components and the elastic method obtains 95% of the energy in the first four components. The difference here can be attributed to the better alignment of the elastic methods as it is able to accurately capture the modes of variation. Kneip and Ramsay's method fails to align the the data well and the components capture more variation from the lack of alignment.

Fig. 4.11a and b presents the first 10 singular values for the simulated data and Berkley growth data, respectively. The blue curve corresponds to performing standard fPCA on the un-aligned data, the orange curve corresponds to the elastic method in Algorithm 4.1, and the green curve corresponds to Kneip and Ramsay's method [37]. Performing standard fPCA on the original data we note that a good portion of the singular values are significant. After performing Algorithm 4.1 we see the amount of energy in the singular values is significantly reduced from the alignment of the data. Our method outperforms Kneip and Ramsay's method in reducing the energy as in both



(a) $\{\tilde{f}_i\}$ (b) $\{\gamma_i^*\}$

Figure 4.9: Alignment results on Berkley Growth data using Kneip and Ramsay's method described in [37].

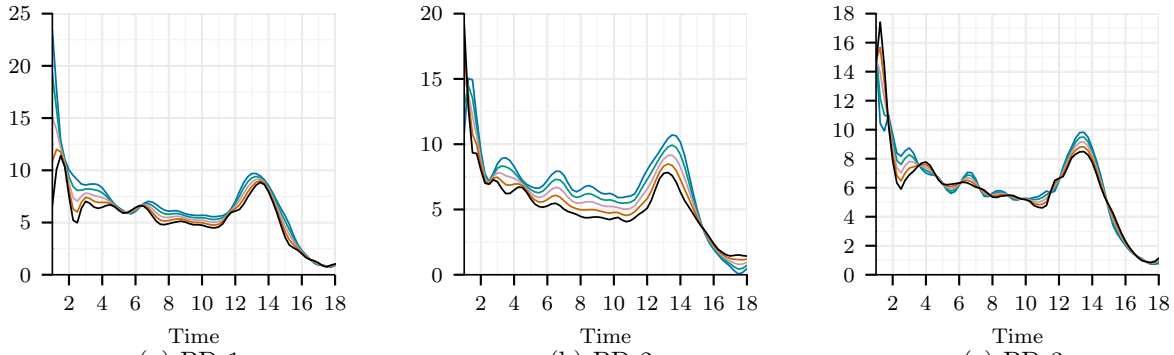
cases the singular values from their method contain more energy than our method. This suggests that we obtain a tighter alignment and more compact representation.

4.2 Functional Partial Least Squares

Functional Partial Least Squares (fPLS) is used to find the fundamental relations between two functions (f and g). Often, it is used in regression, where the variables used as independent variables in the regression analysis display a high degree of correlation; because those variables might be measuring the same characteristics. This problem is known as multi-colinearity, where a high degree of correlation among the predictive variables increases the variance in the estimates of the regression parameters. Therefore, the parameter estimates in a regression model may change with a slight change in data and hence, are not stable for prediction. Comparing with fPCA, which only captures the characteristics of the response function and not the predictive variables, a fPLS model will try to find the direction in the f (response) space that explains the maximum variance direction in the g (predictive) space. In other words, fPLS can be viewed as a covariance maximizer in which we have a pair of observed random functions (f, g). This can be stated as the following maximization problem

$$\{w_f^*, w_g^*\} = \underset{w_f, w_g}{\arg\max} \frac{\mathrm{cov}(\langle f, w_f \rangle, \langle g, w_g \rangle)}{\|w_f\| \|w_g\|}. \tag{4.2.1}$$

Using the same reasons as mentioned previously, we will replace the standard $\mathbb{L}^2$ inner product with the Fisher-Rao inner product due its theoretical properties, and use the SRSF transformation



(a) PD 1 (b) PD 2 (c) PD 3

Figure 4.10: Principal directions on Berkley Growth data using Kneip and Ramsay's method described in [37].

for ease of computation. The benefit of the SRSF transformation is that in the SRSF space the standard $\mathbb{L}^2$ inner product is the Fisher-Rao inner product and is a proper metric and does not exhibit the pinching effect demonstrated in Chapter 2.2.1.

Let $f_1, \cdots, f_n$ be a given set of functions, and $q_{f1}, \cdots, q_{fn}$ be the corresponding SRSFs. Similarly, let $g_1, \cdots, g_n$ be another given set of functions, and $q_{g1}, \cdots, q_{gn}$ be the corresponding SRSFs. The fPLS optimization problem in SRSF space is

$$\{w_{q_f}^*, w_{q_g}^*\} = \underset{w_{q_f}, w_{qg}}{\arg\max} \frac{\text{cov}(\langle q_f, w_{q_f} \rangle, \langle q_g, w_{q_g} \rangle)}{\|w_{q_f}\| \|w_{q_g}\|}. \tag{4.2.2}$$

Since our desire is obtain a warping invariant fPLS we now introduce warping of the SRSF into Eq. 4.2.2. Let $r_f = \langle (q_f, \gamma), w_{q_f} \rangle$ and $r_g = \langle (q_g, \gamma), w_{q_g} \rangle$, then the elastic version of the objective functions is,

$$\{w_{q_f}^*, w_{q_g}^*, \gamma^*\} = \underset{w_{q_f}, w_{q_g}, \gamma}{\arg\max} \left(\frac{E[\max_\gamma (r_f r_g - E[r_f]E[r_g])]}{\|w_{q_f}\| \|w_{q_g}\|} \right). \tag{4.2.3}$$

In practice, where we have an ensemble set functions, the covariance function is replaced with a summation. Let $r_{f,i} = \langle (q_{f,i}, \gamma_i), w_{q_f} \rangle$ and $r_{g,i} = \langle (q_{g,i}, \gamma_i), w_{q_g} \rangle$ be the sample inner products and we can write Eqn. 4.2.3 as

$$\{w_{q_f}^*, w_{q_g}^*, \{\gamma_i^*\}\} = \underset{w_{q_f}, w_{q_g}, \{\gamma_i\}}{\arg\max} \left(\frac{\frac{1}{N} \sum_{i=1}^N (r_{f,i} r_{g,i} - \bar{r}_f \bar{r}_g)}{\|w_{q_f}\| \|w_{q_g}\|} \right) \tag{4.2.4}$$

where $\bar{r}_f = \frac{1}{N} \sum_{i=1}^N r_{f,i}$ and $\bar{r}_g = \frac{1}{N} \sum_{i=1}^N r_{g,i}$ are the means of r_f and r_g, respectively.

Table 4.2: Resulting singular values percentage of cumulative energy on simulated and growth data from Algorithm 4.1 and [37].

(a) Simulated

	1st	2nd	3rd	4th
Elastic Method	40.41	68.26	76.15	80.61
Kneip and Ramsay	86.13	99.88	99.97	99.98

(b) Berkley Growth

	1st	2nd	3rd	4th
Elastic Method	75.39	90.42	93.56	95.37
Kneip and Ramsay	31.66	56.27	63.89	69.87

4.2.1 Optimization over $\{\gamma_i\}$

Next, we focus our attention on solving the objective function in Eqn. 4.2.4 for the warping functions, $\{\gamma_i\}$. We will use the standard gradient ascent method and will represent an element $\gamma \in \Gamma$ by the square-root of its derivative $\psi = \sqrt{\dot{\gamma}}$ as was motivated in Chapter 3.1. The important advantage of this transformation is that since $\|\psi\|^2 = \int_0^1 \psi(t)^2 dt = \int_0^1 \dot{\gamma}(t) dt = \gamma(1) - \gamma(0) = 1$, the set of all such ψs is a Hilbert sphere $\mathbb{S}_\infty$, a unit sphere in the Hilbert space $\mathbb{L}^2$. In other words, the square-root representation simplifies the complicated geometry of Γ to a unit sphere.

By taking this transformation the maximization in Eqn. 4.2.4 over $\{\gamma_i\}$ can be written as

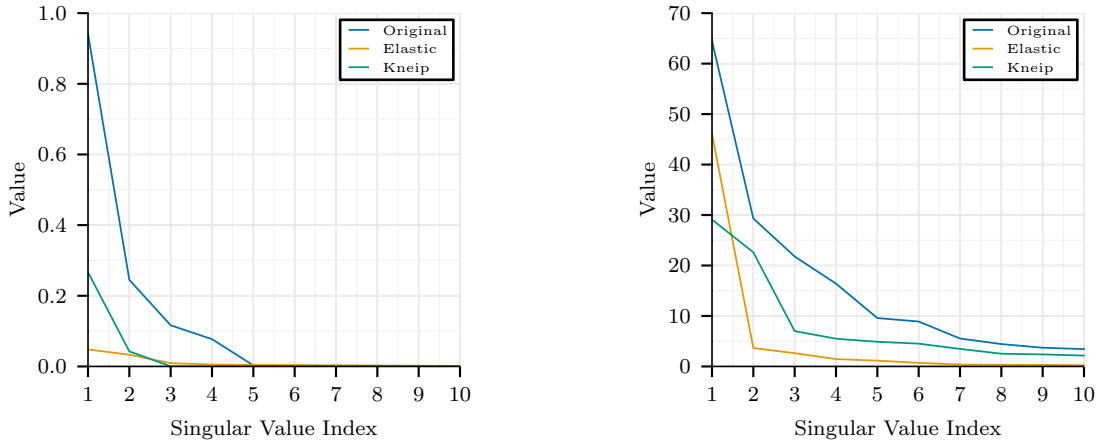
$$H_\gamma = \max_{\{\gamma_i\}} \left[\frac{1}{N} \sum_{i=1}^{N} r_{fi}(\psi_i) r_{gi}(\psi_i) - \frac{1}{N} \sum_{i=1}^{N} r_{fi}(\psi_i) \frac{1}{N} \sum_{j=1}^{N} r_{gj}(\psi_j) \right] \tag{4.2.5}$$

where

$$\begin{aligned} r_{fi}(\psi_i) &= \int_0^1 q_f \left(\int_0^t \psi_i(x)^2 \, dx \right) \psi_i(t) w_{q_f}(t) \, dt \\ r_{gi}(\psi_i) &= \int_0^1 q_g \left(\int_0^t \psi_i(x)^2 \, dx \right) \psi_i(t) w_{q_g}(t) \, dt. \end{aligned}$$

We now wish to take the derivative of Eqn. 4.2.5 with respect to the kth square-root derivative warping function, ψ_k. The first term's derivative is

$$\frac{1}{N} \left(\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k} r_{gk}(\psi_k) + r_{fk}(\psi_k) \frac{\partial r_{gk}(\psi_k)}{\partial \psi_k} \right)$$



(a) Simulated (b) Berkley Growth

Figure 4.11: Resulting singular values on simulated and growth data from standard fPCA, Algorithm 4.1, and [37].

with the remaining terms from the summation being zero. The partial derivatives of r_{fk} and r_{gk} are

$$\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k} = 2\psi_k(t)\int_t^1 \dot{q}_{fk}\left(\int_0^x \psi_k(s)^2\,ds\right)\psi_k(x)w_{q_f}(x)\,dx + q_{fk}\left(\int_0^t \psi_k(s)^2\,ds\right)w_{q_f}(t)$$

$$\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k} = 2\psi_k(t)\int_t^1 \dot{q}_{gk}\left(\int_0^x \psi_k(s)^2\,ds\right)\psi_k(x)w_{q_g}(x)\,dx + q_{gk}\left(\int_0^t \psi_k(s)^2\,ds\right)w_{q_g}(t)$$

The derivative of the second term contains 3 terms: one for when $i = j = k$, one for when $i = k,\ j \neq k$, and $i \neq k, j = k$. The remaining terms from the summations are zero and the derivative is

$$\frac{1}{N^2}\left(\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k}r_{gk}(\psi_k) + r_{fk}(\psi_k)\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k}\right) + \frac{1}{N^2}\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k}\sum_{j\neq k} r_{gj}(\psi_j) + \frac{1}{N^2}\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k}\sum_{i\neq k} r_{fi}(\psi_i)$$

Combining the two derivatives we get the gradient of the objective function as

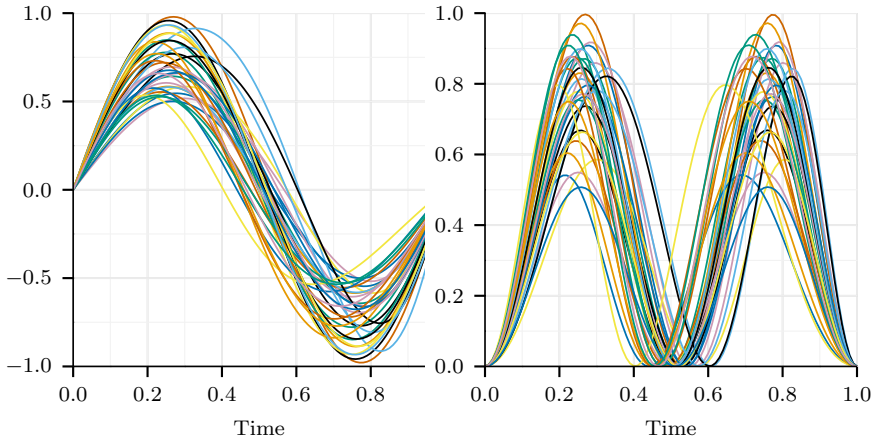
$$\begin{aligned} h(\psi_k) &= \frac{1}{N}\left(\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k} r_{gk}(\psi_k) + r_{fk}(\psi_k)\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k}\right) \\ &\quad -\frac{1}{N^2}\left(\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k} r_{gk}(\psi_k) + r_{fk}(\psi_k)\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k}\right) - \frac{1}{N^2}\frac{\partial r_{fk}(\psi_k)}{\partial \psi_k}\sum_{j\neq k} r_{gj}(\psi_j) \\ &\quad -\frac{1}{N^2}\sum_{i\neq k} r_{fi}(\psi_i)\frac{\partial r_{gk}(\psi_k)}{\partial \psi_k} \end{aligned}$$

We then find the optimal set of $\{\psi_i\}$ and therefore $\{\gamma_i\}$ using gradient descent, as described in Algorithm 4.2.

Algorithm 4.2 Optimization over $\{\gamma_i\}$ for fPLS

1: Set $\psi_i^{(0)} = \psi_{id}$, for $i = 1, \ldots, N$ and set $l = 1$
2: **while** $\|H_\gamma(l+1) - H_\gamma(l)\|^2 < \epsilon$ **do**
3: **for** $i = 1 : N$ **do**
4: Calculate the gradient $h(\psi_i^{(l)})$
5: Find the tangent vector in the direction of h at $\psi_i^{(l)}$ using $h - \left\langle h, \psi_i^{(l)} \right\rangle \psi_i^{(l)}$
6: Update ψ_i component according to $\psi_i^{(l+1)} = \cos(\delta\|h\|)\psi_i^{(l)} + \sin(\delta\|h\|)\frac{h}{\|h\|}$ for a step size $\delta > 0$. This update is simply the exponential map on that sphere at the point $\psi_i^{(l)}$ applied to the tangent vector
7: **end for**
8: Calculate $\gamma_i^* = \int_0^t \psi_i^{l+1}(s)^2\, ds, \quad i = 1, \ldots, N$
9: $l = l + 1$
10: **end while**

A test of the convergence of the algorithm was done using a simulated data set. The functions were constructed according to $f_i = c_i \sin(2\pi t)$ and $g_i = d_i \sin^2(2\pi t)$ where $c_i, d_i \sim U[0.5, 1]$. The functions f_i and g_i were then warped randomly. Since, f_i and g_i consist of only one component we can analytically determine the weight functions w_f and w_g that maximize Eqn. 4.2.4 to be $w_f = \frac{1}{\sqrt{2}}\sin(2\pi t)$ and $w_g = \sqrt{\frac{8}{3}}\sin^2(2\pi t)$. Fig. 4.12 presents the generated functions for f_i and g_i in Panels a and b, respectively. Using this data and the determined weight functions, w_f and w_g, we evaluated Algorithm 4.2 on the simulated data. The algorithm converged in 100 iterations and the cost function is presented in Fig. 4.13.



(a) f_i (b) g_i

Figure 4.12: Simulated f_i and g_i for the testing of Algorithm 4.2.

4.2.2 Full Optimization

Now, we turn our attention to solving the full objective function in Eqn. 4.2.4. First, we will assume that the $\{\gamma_i\}$ are fixed. Our objective function is then

$$H = \max_{w_{q_f}, w_{q_g}} \left(\frac{\frac{1}{N}\sum_{i=1}^{N}(r_{f,i}r_{g,i} - \bar{r}_f\bar{r}_g)}{\|w_{q_f}\|\|w_{q_g}\|} \right),$$

which is a Rayleigh quotient.

The optimal solution of this Rayleigh quotient is the singular-value decomposition (SVD) of the sample covariance matrix. In practice, where q_f and q_g is represented using a finite partition of $[0, 1]$, say with cardinality T, the vectors q_{fi} and q_{gi} have dimension T. We, then, can define a sample covariance operator for r_{fi} and r_{gi} as

$$K = \frac{1}{N-1}\sum_{i=1}^{N}(r_{fi} - \bar{r}_{fi})(r_{gi} - \bar{r}_{gi})^{\mathsf{T}} \tag{4.2.6}$$

Taking the SVD, $K = U\Sigma V^{\mathsf{T}}$ we can calculate w_{q_f} using the first column of U and w_{q_f} as the first column of V. The weight functions w_{q_f} and w_{q_g} can then be converted back to the function space $\mathcal{F}$, via integration, for finding the weight functions of the original functional data.

After we have the weight functions, we will find the warping functions using the dominant or first weight function. Then, after we have found the optimal warping functions, we will find all

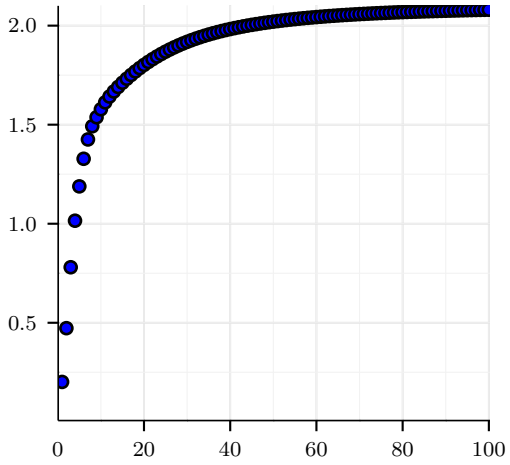


Figure 4.13: Evolution of cost function for Algorithm 4.2.

the components using the aligned data. Algorithm 4.3 presents the overall method for finding the optimal $\{w_{q_{fj}}\}$, $\{w_{q_{gj}}\}$, and $\{\gamma_i\}$.

This procedure results in five items:

1. $\{w_{fi}\}$, the f weight functions,
2. $\{w_{gi}\}$, the g weight functions,
3. $\{\tilde{q}_{fi}\}$, $\{\tilde{q}_{gi}\}$, the two sets of aligned SRSFs,
4. $\{\tilde{f}_i\}$, $\{\tilde{g}_i\}$, the two sets of aligned functions, and
5. $\{\gamma_i^*\}$, the set of optimal warping functions.

4.2.3 Numerical Results

To illustrate the developed elastic fPLS method, we evaluated Algorithm 4.3 on the simulated data described in Section 4.2.1. It was also evaluated on two real data sets, the gait data described in [42] and the iPhone action data set from [44].

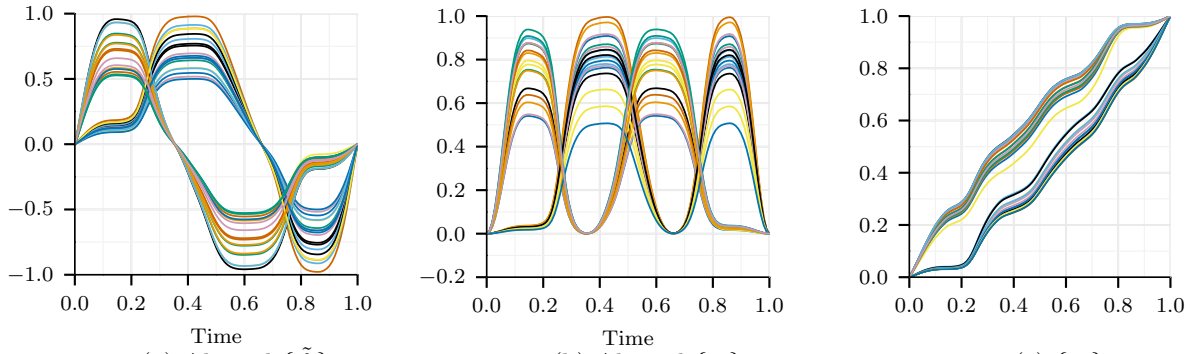
Simulated Pair Data. Using the simulated data set from Section 4.2.1 we demonstrate the results of Algorithm 4.3. The original data for f_i and g_i is presented in Figs. 4.12a and b, respectively. Fig. 4.14 presents the resulting aligned functions and warping functions from

Algorithm 4.3 Functional Partial Least Squares with Warping

1: Initialization Step: Find the first weight functions w_{q_f} and w_{q_g} from the left and right singular vectors from the covariance matrix K, set $l = 1$
2: **while** $\|H^l - H^{l-1}\|^2 < \epsilon$ **do**
3: Find $\{\gamma_i^*\}$ using Algorithm 4.2
4: Compute the aligned SRSFs $\tilde{q}_{fi}$ and $\tilde{q}_{gi}$ using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^{(l)*})\sqrt{\dot{\gamma}_i^{(l)*}}$ for q_{fi} and q_{gi}
5: Calculate w_{q_f} and w_{q_g} from the left and right singular vectors from the covariance matrix K, formed using $\tilde{q}_{fi}$ and $\tilde{q}_{gi}$
6: $l = l + 1$
7: **end while**
8: Form covariance matrix K using final $\tilde{q}_{fi}$ and $\tilde{q}_{gi}$
9: Take singular value decomposition of K
10: Take $\{w_{q_{fj}}\}$ as the n left singular vectors and $\{w_{q_{gj}}\}$ as the n right singular vectors
11: Map the SRSFs, $\{w_{q_{fj}}\}$, and $\{w_{q_{gj}}\}$ back to the function space $\mathcal{F}$ using $\tilde{f}_i(t) = f_i(t_0) + \int_{t_0}^{t} \tilde{q}_i(s)|\tilde{q}_i(s)|\, ds$

Algorithm 4.3. Figs. 4.15a and b presents the resulting weight functions from Algorithm 4.3 for w_f and w_g, respectively.

Figs. 4.15c and d present the resulting weight functions found using the standard fPLS framework for the original f_i and g_i before warping. The standard fPLS framework consists of forming the covariance matrix between f_i and g_i and taking the SVD. The weight functions are then the resulting left and right singular vectors. It should be noted that this framework ignores the phase-variability found in the data. The blue curve in Fig. 4.15 is the first component and the green and pink curves are the next two components. First, the warped functions and warping functions show a clustering of two sets, which actually gives a higher cost function value. The cost function has a value of 0.003 for the original un-warped data. Using our algorithm, the cost function reaches a value of 2.0103. The reason for this is that the data can be warped in the cost function such that it reaches a higher maximal value than that of the original un-warped functions. This is the result of grouping the functions, one set of functions generates positive values for both of the inner-products for f_i and g_i. Similarly, the other half gives negative values for both the inner products of f_i and g_i; when multiplied together is a positive value and therefore the addition generates a higher cost function value.



(a) Aligned $\{\tilde{f}_i\}$ (b) Aligned $\{\tilde{g}_i\}$ (c) $\{\gamma_i\}$

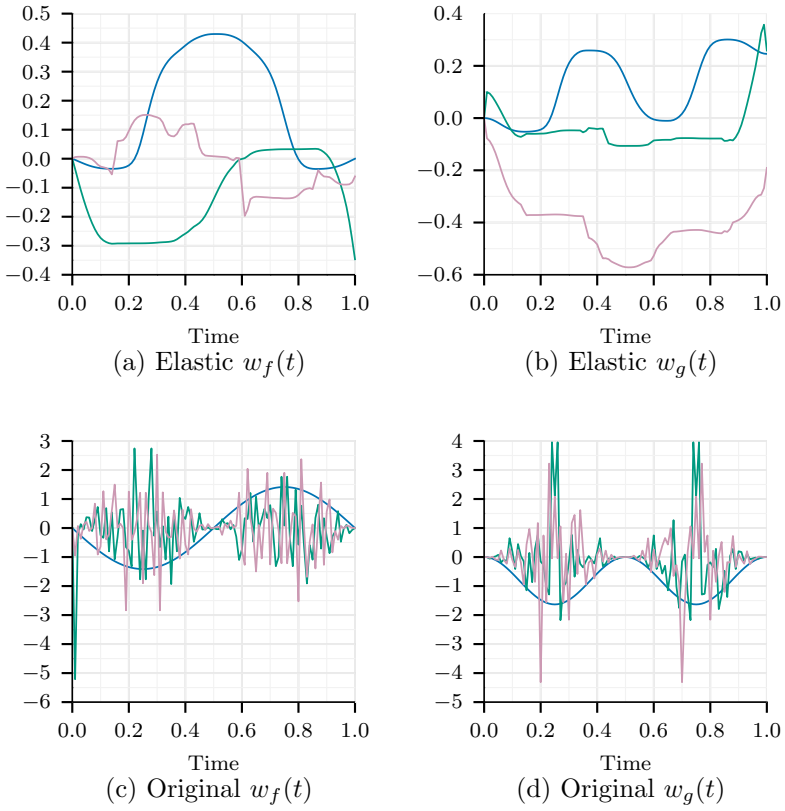
Figure 4.14: The aligned simulated data (a) $\tilde{f}_i$, (b) $\tilde{g}_i$, and corresponding (c) warping functions for the simulated data.

However, our method recovers only one dominant weight function w_f and one dominant weight function w_g, which is expected as our data was constructed with only one basis. This can be seen by first looking at the functions in Fig. 4.15, which for both the elastic weight functions and original weight functions the blue curve has smoothness and regularity. The remaining curves are noise. Moreover, if we examine the resulting singular values in Table 4.3, we see that for the original data and the elastic method we have only one dominant singular value resulting in one dominant weight function. If we look at the singular values resulting from executing the standard fPLS framework on the warped data, we see we have two dominant singular values.

Table 4.3: Resulting singular values on Simulated Data from Algorithm 4.3.

	1st	2nd	3rd	4th
Original Un-Warped Data	3.25e-02	7.75e-18	6.18e-18	5.48e-18
Warped Data	1.01	1.35e-01	1.94e-03	2.07e-04
Elastic Method	201.43	8.30e-02	9.94e-03	6.18e-03

Gait Data. Next, we tested the algorithm on the gait data presented in [42] which has been randomly warped. Fig. 4.16a and b presents the original gait data which consist of the angular rotations in the sagittal plane of the hip and knee of 39 normal 5-year-old children. The observations are taken over a gait cycle consisting of one double step taken by each child, and time is measured



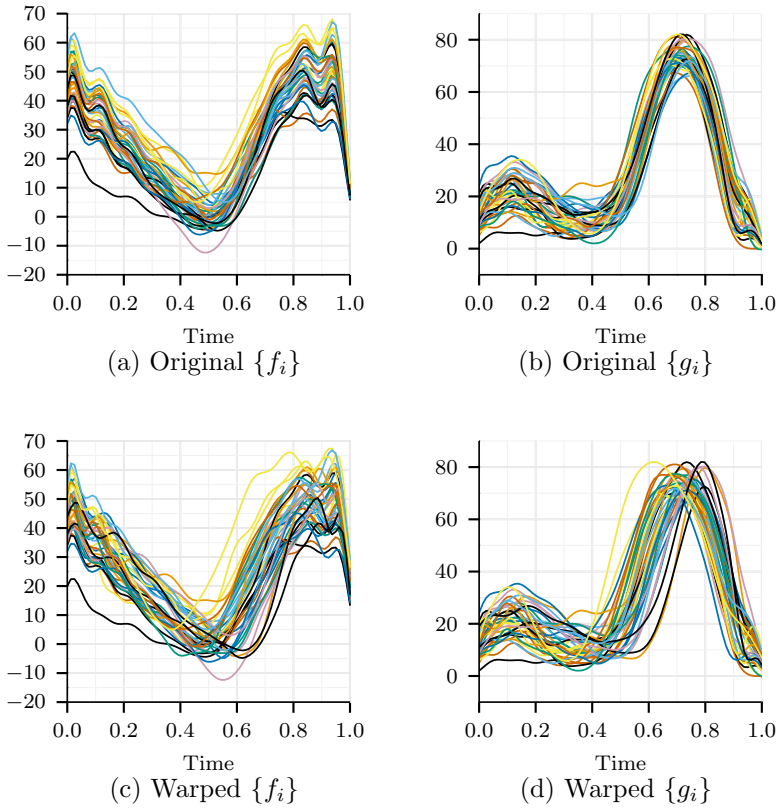
(a) Elastic $w_f(t)$ (b) Elastic $w_g(t)$

(c) Original $w_f(t)$ (d) Original $w_g(t)$

Figure 4.15: The fPLS weight functions (a) w_f and (b) w_g resulting from Algorithm 4.3 and the original weight functions (c) w_f and (d) w_g resulting from standard fPLS on the original un-warped simulated data.

in terms of the cycle. In all cases the cycle has been discretized (mathematically) to a regular grid of 80 points. The original data was then warped randomly to produce phase-variations and Fig. 4.16c and d presents the the randomly warped gait data for the hip and knee, respectively.

Fig. 4.17 presents the resulting aligned functions and warping functions resulting from Algorithm 4.3. Figs. 4.18a and b presents the resulting weight functions from Algorithm 4.3 for w_f and w_g, respectively. As with the simulated data, we see the same two groups of aligned functions which can be attributed to the algorithm achieving a higher objective function value of 91.65 over the original un-warped data achieving 10.73 using the standard fPLS framework. The weight functions resulting from our method are different than those resulting from the standard fPLS framework on the original data (Figs. 4.18c and c). This can be attributed to the two grouping alignment of $\{f_i\}$



(a) Original $\{f_i\}$ (b) Original $\{g_i\}$

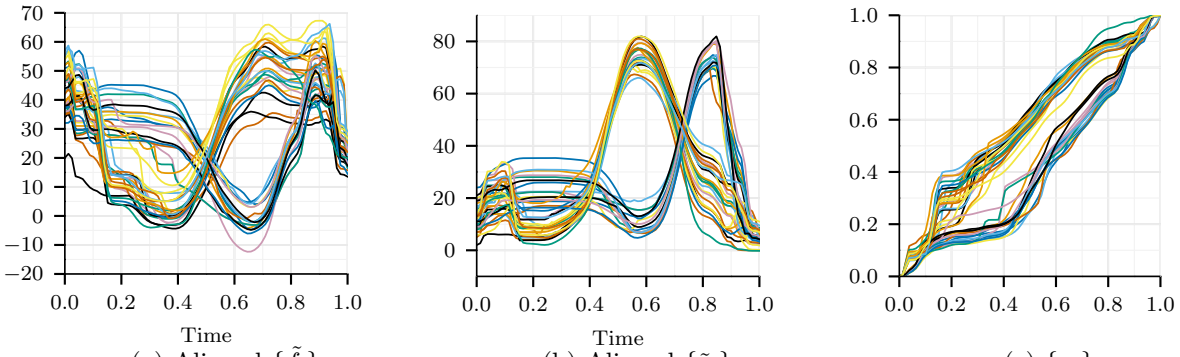
(c) Warped $\{f_i\}$ (d) Warped $\{g_i\}$

Figure 4.16: Original Gait data for the (a) hip and (b) and the randomly warped data for the (c) hip and (d) knee.

and $\{g_i\}$ from our algorithm.

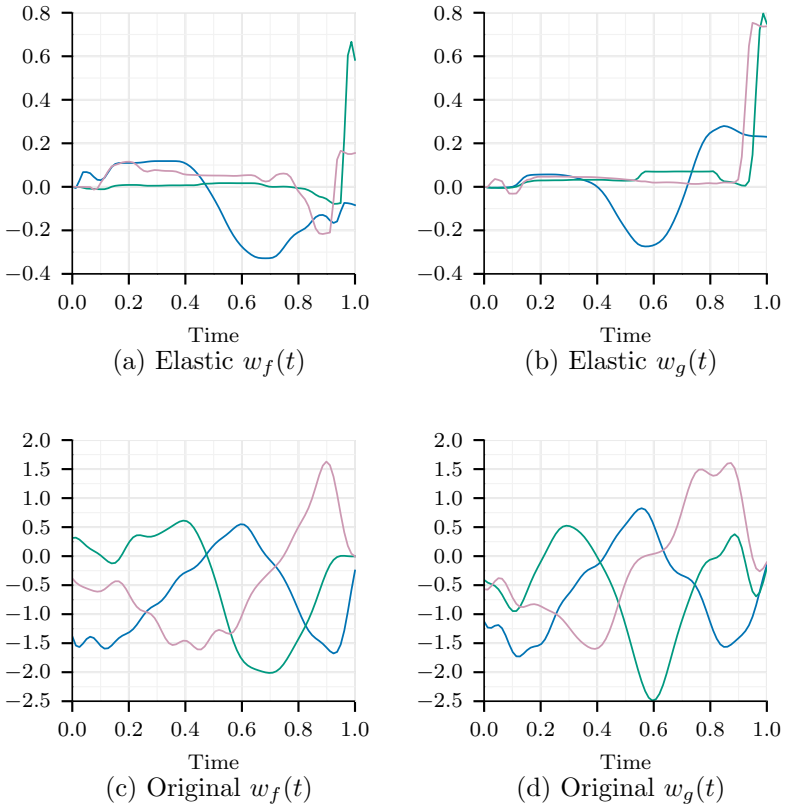
We can also examine the singular values as with the simulated data. Table 4.4 presents the percentage of the cumulative energy for the first four singular values. Specifically, each column represents the percentage of the cumulative energy resulting from the ith singular value. The first four singular values from the standard fPLS framework for the original data are significant and each adds to the energy. This is also similar for the standard fPLS framework for the warped data. Our method packs most of the energy into the first four singular values with the first weight function being dominant. This is from the fact that the warping functions are found using the first component, so the resulting aligned data will have most of its energy in the first component.

iPhone Action Data. Finally, we test our method on the iPhone action data set. This data set consists of aerobic actions recorded from subjects using the Inertial Measurement Unit



(a) Aligned $\{\tilde{f}_i\}$ (b) Aligned $\{\tilde{g}_i\}$ (c) $\{\gamma_i\}$

Figure 4.17: The aligned Gait data (a) $\tilde{f}_i$, (b) $\tilde{g}_i$, and corresponding (c) warping functions.



(a) Elastic $w_f(t)$ (b) Elastic $w_g(t)$

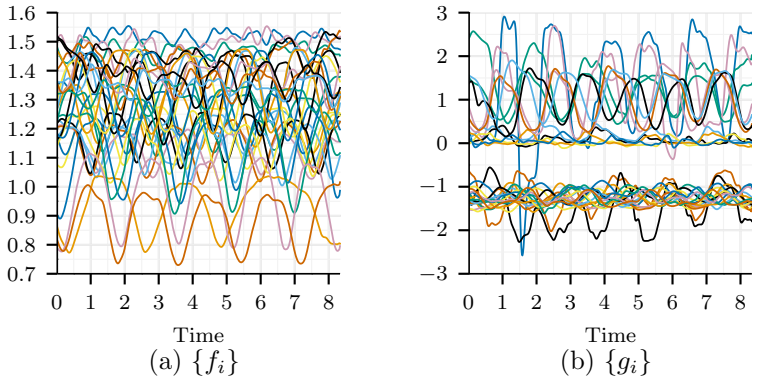
(c) Original $w_f(t)$ (d) Original $w_g(t)$

Figure 4.18: The fPLS weight functions (a) w_f and (b) w_g resulting from Algorithm 4.3 and the original weight functions (c) w_f and (d) w_g resulting from standard fPLS on the original un-warped Gait data.

Table 4.4: Resulting singular values percentage of cumulative energy on Gait data from Algorithm 4.3.

	1st	2nd	3rd	4th
Original Data	23.10	37.59	48.49	55.50
Warped Data	54.78	69.51	75.17	79.06
Elastic Method	88.27	90.63	92.49	93.68

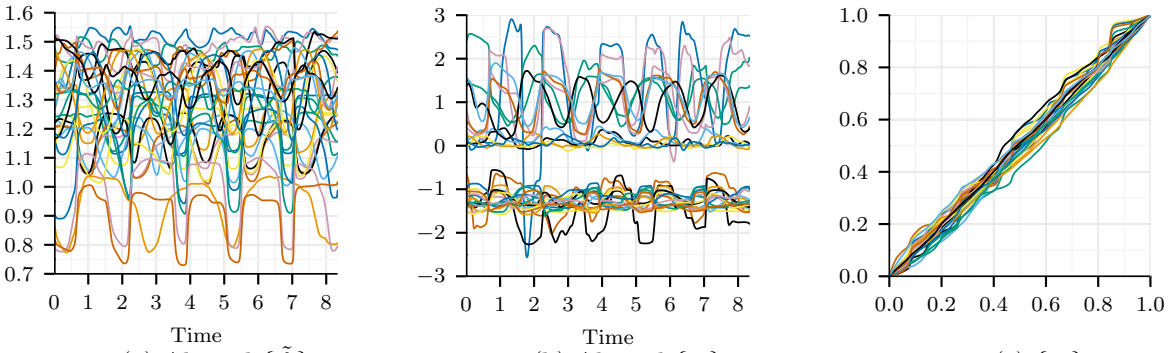
(IMU) on an Apple iPhone 4 smartphone and is described in more detail in Chapter 3.6.2. For our experiments we used the gyrometer data in the x-direction and y-direction for the biking action. To have a robust estimate of the SRSFs $\{q_{fi}\}$ and $\{q_{gi}\}$, we first smooth the original signals 25 times, $\{f_i\}$ and $\{g_i\}$, using the standard box filter described in Chapter 3.6.1. Figs. 4.19a and b present the original data for the x-gyrometer and y-gyrometer, respectively.



(a) $\{f_i\}$ (b) $\{g_i\}$

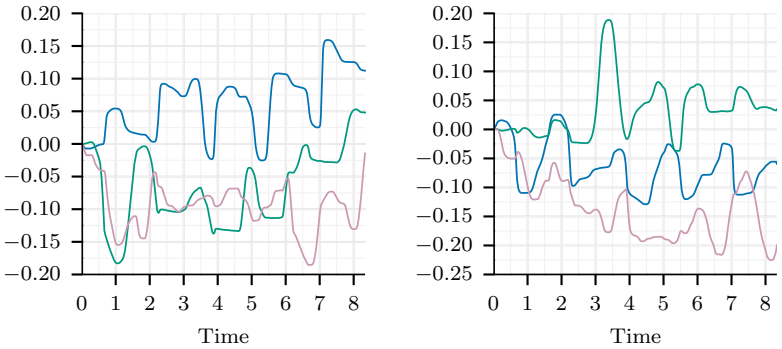
Figure 4.19: Original iPhone bike data for the (a) x-gyrometer and (b) y-gyrometer.

Fig. 4.20 presents the aligned x- and y-gyrometer and corresponding warping functions in Panels a, b, and c, respectively. Comparing to the original functions, there is is some shifting of peaks with a most of the alignment being linear shifts. The warping functions in Panel c demonstrate the small amount of warping that was required for the alignment as most of the warping functions are close to γ_{id}. The corresponding weight functions w_f and w_g are presented in Fig. 4.21a and b, respectively. All three components for both w_f and w_g showed significant singular values. As with the previous data sets, the objective function from our algorithm achieves a higher value of 0.74 compared to the standard fPLS framework on the original data of 0.32.



(a) Aligned $\{\tilde{f}_i\}$ (b) Aligned $\{\tilde{g}_i\}$ (c) $\{\gamma_i\}$

Figure 4.20: The aligned iPhone action data (a) $\tilde{f}_i$, (b) $\tilde{g}_i$, and corresponding (c) warping functions.



(a) Elastic $w_f(t)$ (b) Elastic $w_g(t)$

Figure 4.21: The fPLS weight functions (a) w_f and (b) w_g resulting from Algorithm 4.3 for iPhone action data.

CHAPTER 5

JOINT ALIGNMENT AND FUNCTIONAL REGRESSION

This chapter discusses the problem of *elastic* regression analysis, in which we incorporate phase-variability in a non-parametric way into the functional linear regression model. This alters how the model parameters are identified as the warping is now an additional parameter in the model. As with the elastic component analysis, this approach is more natural and creates more parsimonious regression models for data that contains phase-variability.

As in the previous chapter we assume that we have the functions and their corresponding square-root slope functions (SRSFs) as described in the Chapter 2. We have three goals in this chapter: First is to develop the elastic functional linear regression model. Second, we derive the functional logistic regression model and the multinomial logistic model from the linear model. Finally, we demonstrate the power of the elastic models in classification of physiological data.

5.1 Elastic Functional Linear Regression

One of the commonly studied problems in functional data analysis is functional linear regression. More precisely, let the predictor functions be given by $\{f_i : [0,T] \to \mathbb{R},\ i = 1, 2, \ldots, n\}$ and the corresponding response variables be y_i. The standard functional linear regression model for this set of observations is

$$y_i = \alpha + \int_0^T f_i(t)\beta(t)\,dt + \epsilon_i, \quad i = 1, \ldots, n \tag{5.1.1}$$

where $\alpha \in \mathbb{R}$ is the intercept, $\beta(t)$ is the regression-coefficient function and $\epsilon_i \in \mathbb{R}$ are random errors. The model parameters are usually estimated by minimizing the sum of squared errors (SSE),

$$\{\alpha^*, \beta^*(t)\} = \underset{\alpha,\beta(t)}{\arg\min} \sum_{i=1}^{n} |y_i - \alpha - \int_0^T f_i(t)\beta(t)\,dt|^2.$$

These values form maximum-likelihood estimators of parameters under the additive white-Gaussian noise model. As stated in Chapter 2 one of the problems with this approach, is that since $\beta(t)$

is infinite dimensional, we have infinite degrees of freedom to form $\beta(t)$ in which we can make the SSE equal zero. Ramsay [52] proposed to represent $\beta(t)$ using p basis functions in which p is hopefully large enough to capture all variations of $\beta(t)$ or use a penalty term which shrinks the variability of $\beta(t)$ or smooths it response. However, this model assumes the data is aligned and has no phase-variability.

To define a model that contains phase-variation we use previous notation, however to remind the reader: Let f be a absolutely continuous real-valued function on $[0,1]$ and $\mathcal{F}$ denotes the set of all such functions. Also, let Γ be the set of boundary-preserving diffeomorphisms of the unit interval $[0,1]$, i.e. $\Gamma = \{\gamma : [0,1] \to [0,1] |\ \gamma(0) = 0,\ \gamma(1) = 1, \gamma \text{ is a diffeomorphism}\}$. Γ is a group with the action of composition and the identity element given by $\gamma_{id}(t) = t$. Its elements play the role of warping functions. For any $f \in \mathcal{F}$ and $\gamma \in \Gamma$, the composition $f \circ \gamma$ denotes the time-warping of f by γ.

Incorporating time warping into the functional regression model in Eqn. 5.1.1 gives us a predictive variable that is defined as

$$\begin{aligned} y_i^0 &= \alpha + \int_0^1 f_i^0(t)\beta(t)\,dt \\ y_i &= y_i^0 + \epsilon_i, \quad i = 1, \ldots, n. \end{aligned} \tag{5.1.2}$$

The predictive functions are then observed under phase-variability

$$f_i = f_i^0 \circ \gamma_i, \tag{5.1.3}$$

and we will identify this model by minimizing the SSE with the observed pair (y_i, f_i). The estimation problem now includes maximization of the likelihood over the warping functions, in addition to the quantities α and β:

$$\{\alpha^*, \beta^*(t), \{\gamma_i^*\}\} = \underset{\alpha, \beta(t), \{\gamma_i\}}{\arg\min} \sum_{i=1}^{n} |y_i - \alpha - \int_0^1 (f_i \circ \gamma_i)(t)\beta(t)\,dt|^2. \tag{5.1.4}$$

However, Chapter 2.2.2 demonstrated that the computation of optimal time warping in the original function space has some fundamental and practical issues. These include the pinching effect (singularity in matching) and non-symmetric nature of $\mathbb{L}^2$ norm between the warped functions.

Specifically, if the functions $f_i(t)$, $i = 1, \ldots, n$ are constructed using $f_i(t) = c_i f(t)$ where $c_i > 0$ the solution to the cost function in Eqn. 5.1.4 becomes degenerate. For example we can construct

functions f_i using

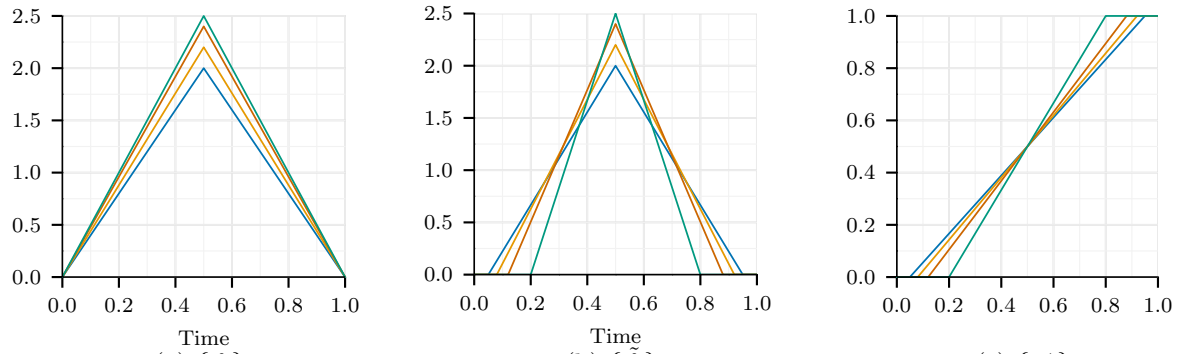
$$f(t) = \begin{cases} 4t & 0 < t < 1/2 \\ 4 - 4t & 1/2 \leq t \leq 1 \end{cases}$$

and if we assume $\alpha = 0$ then

$$\begin{aligned} y_i &= \int_0^1 (f_i \circ \gamma_i)(t)\beta(t)\, dt \\ &= \int_0^1 c_i (f \circ \gamma_i)(t)\beta(t)\, dt. \end{aligned}$$

An optimal solution can be found such that $\beta(t)$ would be a constant β and we can find a γ_i such that $\frac{y_i}{c_i \beta} = \int_0^1 f(\gamma_i(t))dt$. Since, $\beta(t)$ is a constant it has no meaning in the regression model and the solution is therefore degenerate.

A simple example of this problem is shown in Fig. 5.1 where the original functions are constructed as described above using $c_i \in \{1, 1.1, 1.2, 1.25\}$. The original functions, aligned functions $\{f_i \circ \gamma_i\}$, and optimal warping functions $\{\gamma_i^*\}$ are shown in Panels a, b, and c, respectively. In this case the warping functions pinch the functions to obtain a SSE value of 0 forming a degenerate solution.



(a) $\{f_i\}$ (b) $\{\tilde{f}_i\}$ (c) $\{\gamma_i^*\}$

Figure 5.1: Example of pinching problem in functional linear regression with warping.

Additionally, from Chapter 2.2.1 we know that $\|f \circ \gamma\| \neq \|f\|$. In other words we can make $\|f \circ \gamma\|$ any value we want in the range of f just by selecting a γ to select that point. To address these and related problems, we introduced a mathematical expression for representing a function. This function, $q : [0, 1] \to \mathbb{R}$, is called the *square-root slope function* or SRSF of f, and is defined

in the following form:

$$q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}\ .$$

With this transformation we now have $\|q \circ \gamma\| = \|q\|$ which preserves the energy of q. We suggest an alternative data model where we use SRSFs $\{q_i\}$, rather than $\{f_i\}$ themselves as predictors. The rest of the model remains same and in SRSF space is defined to be

$$\begin{aligned} y_i^0 &= \alpha + \int_0^1 q_i^0(t)\beta(t)\,dt \\ y_i &= y_i^0 + \epsilon_i, \quad i = 1, \ldots, n. \end{aligned} \tag{5.1.5}$$

The predictive functions are then observed under phase-variability

$$q_i = (q_i^0, \gamma_i), \tag{5.1.6}$$

and we will identify this model by minimizing the SSE with the observed pair (y_i, q_i), where the expectation is over q. That is,

$$\{\alpha^*, \beta^*(t), \{\gamma_i^*\}\} = \underset{\alpha,\beta(t),\{\gamma_i\}}{\arg\min} \sum_{i=1}^{n} |y_i - \alpha - \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt|^2. \tag{5.1.7}$$

It should be mentioned that if $\hat{\beta}$ is a minimizer of the cost function, then so is $\hat{\beta} \circ \gamma$ for any $\gamma \in \Gamma$ since the cost function is invariant to random warpings of its input variables. So, we have an extra degree of freedom in choosing an arbitrary element of the set $\{\hat{\beta} \circ \gamma | \gamma \in \Gamma\}$. To make this choice unique, we can define a special element of this class as follows. Let $\{\gamma_i^*\}$ denote the set of optimal warping functions, one for each i, in Eqn. 5.1.7. Then, we can choose the $\hat{\beta}$ to that element of its class such that the mean of $\{\gamma_i^*\}$, denoted by γ_μ, is γ_{id}, the identity element. (The notion of the mean of warping functions and its computation are described in Algorithm 3.1.)

5.1.1 Maximum-Likelihood Estimation Procedure

In the elastic model, we need to estimate α^*, β^*, and $\{\gamma_i^*\}$ using Eqn. 5.1.7. Note that the optimization is of significant challenge because α and $\beta(t)$ are parameters in the regression and γ_i is the time warping in each observation. In this dissertation, we propose an iterative procedure to update them alternatively.

Optimization over warping functions. First, we will assume that α and $\beta(t)$ are given and our goal is to solve for the set of optimal warping functions, $\{\gamma_i^*\}$. Since the summation in Eqn. 5.1.7 is over i, we can estimate the elements of the set $\{\gamma_i\}$ individually for each i. Specifically, we can move the minimization over γ_i inside the summation and each γ_i^* can be found by solving

$$\gamma_i^* = \arg\min_{\gamma_i} |y_i - \alpha - \int_0^1 (q_i, \gamma_i)(t)\beta(t)\, dt|^2. \tag{5.1.8}$$

To first the solve the minimization in Eqn. 5.1.8, we can easily get

$$\begin{aligned} \gamma_m &= \arg\min_{\gamma_i} \int (q_i, \gamma_i)(t)\beta(t)\, dt = \arg\min_{\gamma_i} \|(q_i, \gamma_i) + \beta\|^2 \\ \gamma_M &= \arg\max_{\gamma_i} \int (q_i, \gamma_i)(t)\beta(t)\, dt = \arg\min_{\gamma_i} \|(q_i, \gamma_i) - \beta\|^2. \end{aligned}$$

The solution to these two optimization problems comes from the dynamic programming algorithm [3]. Next, let $y_m = \int (q_i, \gamma_m)(t)\beta(t)\, dt$ and $y_M = \int (q_i, \gamma_M)(t)\beta(t)\, dt$, we can find the optimal γ_i^* using the following algorithm:

Algorithm 5.1 Optimization over γ_i for Elastic Functional Linear Regression

1: **if** $y_i > \alpha + y_M$ **then**
2: $\quad \gamma_i^* = \gamma_M$
3: **end if**
4: **if** $y_i < \alpha + y_m$ **then**
5: $\quad \gamma_i^* = \gamma_m$
6: **end if**
7: **if** $\alpha + y_m < y_i < \alpha + y_M$ **then**
8: $\quad$ Look for the optimal warping function in the following form:

$$\gamma = s\gamma_M + (1-s)\gamma_m, \quad s \in [0,1]$$

9: $\quad$ Let $f(\gamma) = y_i - \alpha - \int (q_i, \gamma)(t)\beta(t)\, dt$ and $g(s) = f(s\gamma_M + (1-s)\gamma_m)$
10: $\quad$ Then $g(0) = f(\gamma_m) > 0$, and $g(1) = f(\gamma_M) < 0$ and based on the intermediate value theorem, we can use a secant-type method for finding the optimal s^*, such that $g(s^*) = 0$
11: **end if**

Optimization over α and β. In this step we assume that the $\{\gamma_i\}$ are fixed, and adopt a conventional basis-based approach for estimating α^* and β^*. The coefficient function $\beta(t)$ is represented by a set of basis functions (such as the Fourier basis or a B-spline basis) [θ_i, $i =$

$1, \ldots, p]$, and takes the form $\beta(t) = \sum_{i=1}^{p} b_i \theta_i$, where $b_i \in \mathbb{R}$ is the coefficient. We can combine all the parameters – intercept α and coefficients b_is – in a vector form $\mathbf{b} = [\alpha, b_1, \ldots, b_p]^\mathsf{T}$ and $\mathbf{y} = [y_1, \ldots, y_n]^\mathsf{T}$. The regression model then can be written as,

$$\mathbf{y} = Z^\mathsf{T}\mathbf{b} + \epsilon_i, \tag{5.1.9}$$

where $Z = [\mathbf{1}\ \Theta]$ and the (i,j)th entry in Θ is $\int (q_i, \gamma_i)(t)\theta_j(t)\, dt$. We then can estimate optimal parameter vector $\mathbf{b}^*$ the using ordinary least squares,

$$\mathbf{b}^* = (Z^\mathsf{T}Z)^{-1}Z^\mathsf{T}\mathbf{y} \tag{5.1.10}$$

and the optimal $\beta^*(t) = \sum_{i=1}^{p} b_i^* \theta_i$.

To perform the overall minimization we alternate between finding the optimal warping functions $\{\gamma_i\}$ and finding the optimal $\mathbf{b}$. The algorithm for computing optimal α, β, and $\{\gamma_i\}$ is given in Algorithm 5.2.

Algorithm 5.2 Elastic Functional Linear Regression

1: Initialization Step: set $\{\gamma_i\} = \gamma_{id}$, calculate SRSFs $\{q_i\}$, set $l = 1$.
2: **while** $\|H^{(l+1)} - H^{(l)}\|^2 < \epsilon$ **do**
3: Find $\alpha^{(l)}$ and $\beta^{(l)}(t)$ using basis-based OLS
4: Find $\{\gamma_i\}^{(l)}$ using Algorithm 5.1
5: Calculate SSE $H^{(l)} = \sum_{i=1}^{n} |y_i - \alpha^{(l)} - \int (q_i, \gamma_i^{(l)})(t)\beta^{(l)}(t)\, dt|^2$
6: $l = l + 1$
7: **end while**
8: Find the mean (γ_μ) of $\{\gamma_i^*\}$ using Algorithm 3.1
9: Update $\gamma_i^* \mapsto \gamma_i^* \circ \gamma_\mu^{-1}$
10: Update $\beta^{(l)} = (\beta^{(l)} \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma_\mu^{-1}}}$
11: Compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma_i^*}}$.
12: Map the SRSFs, $\beta(t)$, and $\tilde{q}_i$ back to the function space $\mathcal{F}$ using $\tilde{f}_i(t) = f_i(t_0) + \int_{t_0}^{t} \tilde{q}_i(s)|\tilde{q}_i(s)|\, ds$

This procedure results in four items:

1. α^*, optimal α
2. $\beta^*(t)$, optimal regression function,
3. $\{\gamma_i^*\}$, the set of optimal warping functions, and
4. $\{\tilde{f}_i\}$, the set of aligned functions.

5.1.2 Prediction

After the model has been identified, the next question that arises is how the prediction is performed. The answer is simple if the test data has not been observed with phase variation as prediction would be computed using

$$y_i = \alpha + \int_0^1 q_i(t)\beta(t)\,dt. \tag{5.1.11}$$

However, a problem arises when the input has been observed with phase variation. The prediction would be performed using

$$y_i = \alpha + \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt, \tag{5.1.12}$$

but the question remains on what γ_i should be.

To solve this problem, we propose the following: 1) Take the observed SRSF, q_i and find the SRSF from the training set which has the smallest distance using $\|q_i - q_{train}\|^2$. 2) Determine the γ_i to be used for prediction as the corresponding γ_i for the closest training sample. 3) Find the predictive value, y_i, using Eqn. 5.1.12 with the corresponding chosen γ_i.

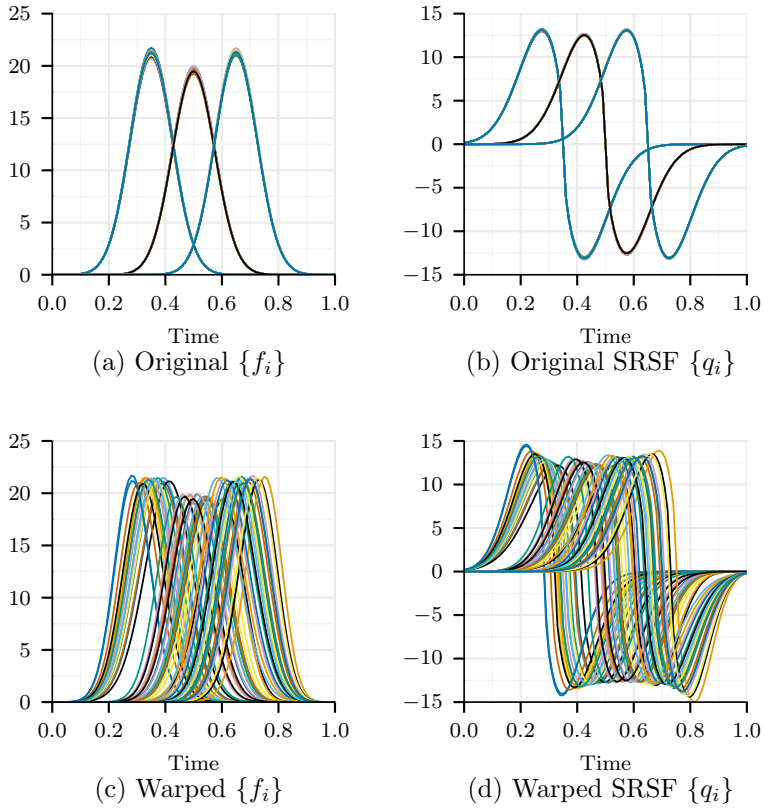
5.1.3 Experimental Results

To illustrate the developed elastic functional linear regression method we evaluated Algorithm 5.2 on a simulated data constructed using

$$f_i(t) = a_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu_j)^2}{2\sigma^2}\right),$$

where $a_i \sim \mathcal{N}(d_j, 0.05)$ and $d_j \in \{4, 3.7, 4\}$. The mean was chosen according to $\mu_j \in \{0.35, 0.5, 0.65\}$ and 20 functions were generated for each μ_j and $\sigma = 0.075$. The generated functions are shown in Fig. 5.2a with corresponding SRSFs in Fig. 5.2b. The functions were then randomly warped to generate the warped data, $\{f_i\}$ and $\{q_i\}$, presented in Figs 5.2c and d, respectively. The response variable y_i was generated with $\alpha = 0$, $\beta(t) = 0.5\sin(2\pi t) + 0.9\cos(2\pi t)$, and the original SRSFs.

The resulting estimated $\beta(t)$ from Algorithm 5.2 is presented in Fig. 5.3. We compared the results to the estimated $\beta(t)$ using standard function linear regression (FLR) found in the literature [52] on the the original and warped data. In the figure, the blue curve corresponds to running standard FLR on the original data which is the true $\beta(t)$, the orange curve is performing standard FLR on the warped data, and the red curve is the estimated $\beta(t)$ using the elastic method. Both



(a) Original $\{f_i\}$ (b) Original SRSF $\{q_i\}$

(c) Warped $\{f_i\}$ (d) Warped SRSF $\{q_i\}$

Figure 5.2: Original Simulated data in (a) f space and (b) SRSF space with corresponding warped data in (c) f space and (d) SRSF space.

the red and orange curves differ from the truth, however, the $\beta(t)$ found using the elastic method is closer to the the true shape of the original $\beta(t)$. Fig. 5.4 presents the resulting aligned functions, aligned SRSFs, and corresponding estimated warping functions from Algorithm 5.2 in Panels a, b, and c, respectively. The functions are aligned and clustered into three distinct groups which is similar to how the original data was constructed. Fig. 5.5 presents the evolution of the sum of squared errors (SSE) for the algorithm which converged in 3 iterations and the final value being nearly zero at 2.83e-10. The SSE for standard FLR on the original data is a little higher at 0.01 and the SSE for standard FLR on the warped data was 13.94. With the additional degree of freedom of γ_i we are able to drive the SSE nearly to zero.

In order to more accurately test the performance of Algorithm 5.2 we performed a 5-fold cross-validation experiment to evaluate the prediction error of the elastic method versus the standard

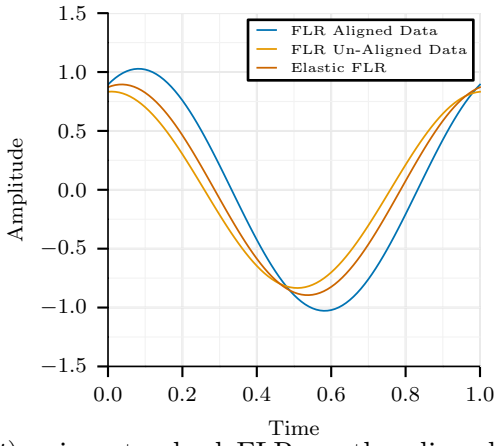
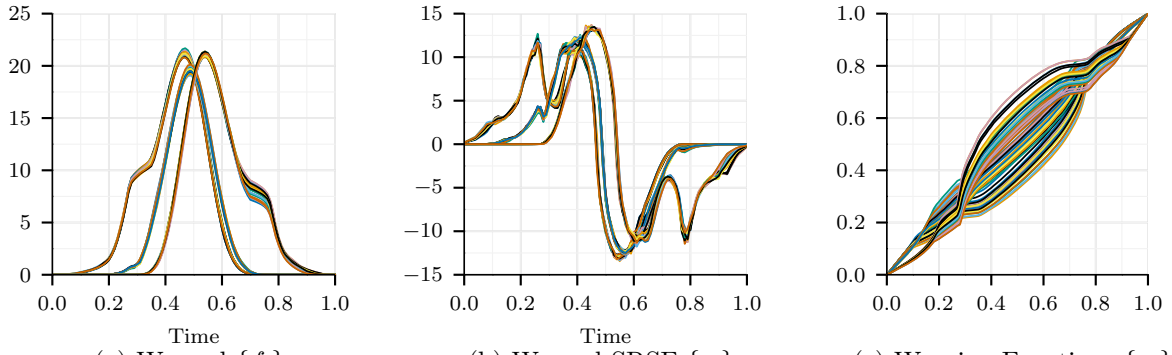


Figure 5.3: Estimated $\beta(t)$ using standard FLR on the aligned and un-aligned data and using Elastic FLR on the un-aligned data.

FLR. Prediction was performed as was described in Section 5.1.2 and since we have the true $\beta(t)$ and α we can calculate the true y for comparison. Table 5.1 presents the mean prediction error across the folds with the corresponding standard error in parentheses. In this table, we compare the elastic FLR with four methods: 1) standard FLR on the original data, 2) standard FLR on the warped data, 3) pre-align FLR, which pre-aligns the training data using Algorithm 2.1 and performs standard FLR. Prediction is performed by taking the warped sample and aligning it to the mean function found using the alignment; and calculating the predicted value using the identified model. Lastly, 4) cluster FLR, which clusters the training response data, $\{y_i\}$, using k-means, and then aligns the training data inside each cluster using the same alignment algorithm as the third method. Prediction is performed by taking the warped sample and finding the cluster it is closest to in the training data. Then align it to the mean function of the corresponding cluster, and then calculate the predicted value. Methods 3) and 4) are generalizations of most suggested techniques when the data is not aligned prior to performing regression, where the described methods suggest a "pre-alignment".

Comparing with the standard FLR on the original data, which is the truth in this case, we see that the elastic method obtains a lower MSE compared to running the standard methods on the warped data. Specifically, the pre- alignment method obtains the highest MSE, which is attributed to the pre-alignment destroying the structure of the underlying model. The clustering method is comparable to performing just standard FLR on the un-aligned data. However, it demonstrates a



(a) Warped $\{f_i\}$ (b) Warped SRSF $\{q_i\}$ (c) Warping Functions $\{\gamma_i\}$

Figure 5.4: Warped Simulated data in (a) $\mathcal{F}$ space and (b) SRSF space with corresponding (c) warping functions.

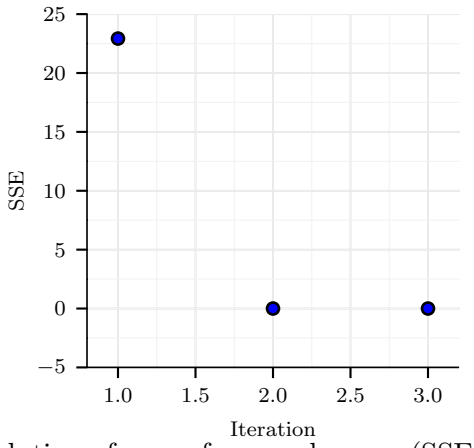


Figure 5.5: Evolution of sum of squared errors (SSE) for Algorithm 5.2.

larger standard deviation. The higher standard deviation results from some of the functions being assigned to a incorrect cluster. The difficulty of the problem is increased if the number of clusters is not known *a-priori*. Therefore, the elastic method is able to remove most of the variability of γ and obtain the lowest prediction error over current methods.

5.2 Elastic Functional Logistic Regression

It is common in some situations to have functional data where the response variable is binary, $y_i \in \{-1, 1\}$, for $i = 1, \ldots, n$. In this case one would want to classify the samples to a specific class

Table 5.1: Mean prediction error using 5-fold cross-validation and standard deviation in parentheses for simulated data using functional regression.

	MSE (SE)
FLR Original Data	0.00416 (0.00057)
FLR Warped Data	1.327 (0.636)
Pre-Align FLR	39.540 (4.454)
Cluster FLR	1.465 (1.798)
Elastic FLR	**0.436 (0.126)**

given the functional predictor. We define the probability of the function f_i being in class 1 ($y_i = 1$) as

$$P(y_i = 1|f_i) = \frac{1}{1 + \exp\left(-\left[\alpha + \int_0^1 f_i(t)\beta(t)\,dt\right]\right)}.$$

This is nothing but the logistic link function $\phi(t) = 1/(1 + \exp(-t))$ applied to the conditional mean in a linear regression model: $\alpha + \int_0^1 f_i(t)\beta(t)dt$ [23]. Using this relation, and the fact that $P(y = -1|f_i) = 1 - P(y = 1|f_i)$, we can express the data likelihood as:

$$\pi(\{y_i\}|\{f_i\}, \alpha, \beta) = \prod_{i=1}^{n} \frac{1}{1 + \exp\left(-y_i\left[\alpha + \int_0^1 f_i(t)\beta(t)\,dt\right]\right)}.$$

Assuming we observe a sequence of i.i.d. pairs $\{f_i(t), y_i\}, i = 1, \cdots, n$, the model is identified by maximizing the log-likelihood according to,

$$\{\alpha^*, \beta^*\} = \arg\max_{\alpha,\beta(t)} \left(\log \pi(\{y_i\}|\{f_i\}, \alpha, \beta)\right).$$

This optimization has been the main focus of the current literature, see e.g., [10, 21, 52].

As with functional linear regression, this model assumes that the functions, f_i, are aligned and have no phase-variability. If we incorporate time warping into the model using the SRSF framework as motivated earlier we get the following probability of q_i (the SRSF of f_i) being in class 1 as

$$P(y = 1|q_i) = \frac{1}{1 + \exp\left(-\left[\alpha + \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt\right]\right)}.$$

For a set of given observation $(f_i(t), y_i), i = 1, \cdots, n$, the maximum likelihood problem is given by:

$$\begin{aligned}\{\alpha^*, \beta^*(t), \{\gamma_i^*\}\} &= \arg\max_{\alpha,\beta(t),\{\gamma_i\}} \sum_{i=1}^{n} \log\left(\phi\left(y_i\left[\alpha + \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt\right]\right)\right) \\ &= \arg\max_{\alpha,\beta(t)} \sum_{i=1}^{n}\left(\arg\max_{\gamma_i} \log\left(\phi\left(y_i\left[\alpha + \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt\right]\right)\right)\right) \quad (5.2.1)\end{aligned}$$

5.2.1 Maximum-Likelihood Estimation Procedure

In the elastic model, we need to estimate α^*, β^*, and $\{\gamma_i^*\}$ using Eqn. 5.2.1. Note that the optimization is of significant challenge because α and $\beta(t)$ are parameters in the regression and γ_i is the time warping in each observation. Again, we propose an iterative procedure to update them alternatively.

Optimization over warping functions. First, we will assume that α and β are fixed and our goal is to find the set of optimal warping functions, $\{\gamma_i^*\}$. As indicated in the second part of Eqn. 5.2.1, we can estimate each γ_i separately by solving

$$\begin{aligned}\gamma_i^* &= \arg\max_{\gamma_i} \log\left(\phi\left(y_i\left[\alpha + \int_0^1 (q_i,\gamma_i)(t)\beta(t)\,dt\right]\right)\right) \\ &= \arg\max_{\gamma_i}\left(y_i\left[\int_0^1 (q_i,\gamma_i)(t)\beta(t)\,dt\right]\right).\end{aligned}$$

Note that since $\|(q_i,\gamma_i)\| = \|q_i\|$, we can find the optimal γ_i using

$$\gamma_i^* = \arg\max_{\gamma_i} \int_0^1 (q_i,\gamma_i)(t) y_i \beta(t)\,dt = \arg\min_{\gamma_i} \|(q_i,\gamma_i) - y_i\beta\|^2. \tag{5.2.2}$$

That is, γ_i^* is the optimal warping of the function $q_i(t)$ to match the function $y_i\beta(t)$. The solution can be effectively computed using a dynamic programming algorithm on a finite grid [3].

Optimization over α and β. In this step we assume that the $\{\gamma_i\}$ are fixed, and adopt a conventional basis-based approach for estimating α^* and β^* as in the linear case. Let, $\beta(t) = \sum_{i=1}^p b_i\theta_i$, where θ_i is the ith basis function and b_i is the corresponding coefficient. We can combine all the parameters – intercept α and coefficients b_is – in a vector form $\mathbf{b} = [\alpha, b_1, \ldots, b_p]^\mathsf{T}$. Let $\mathbf{z}_i = [1, \int_0^1 (q_i,\gamma_i)(t)\theta_1(t)\,dt, \ldots, \int_0^1 (q_i,\gamma_i)(t)\theta_p(t)\,dt]^\mathsf{T}$. Based on Eqn. 5.2.1, the optimal parameter vector is given as follows:

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathbb{R}^{p+1}} \sum_{i=1}^n \log\left(\phi\left(y_i\mathbf{b}^\mathsf{T}\mathbf{z}_i\right)\right), \tag{5.2.3}$$

There is no analytical solution to this optimization problem. Since the objective function is concave, we can use a numerical method such as Conjugate Gradient or the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [46]. To use these algorithms, we need the gradient of the log-likelihood, L, which is given by:

$$\nabla L(\mathbf{b}) = \sum_{i=1}^n -y_i\mathbf{z}_i(\phi(y_i\mathbf{b}^\mathsf{T}\mathbf{z}_i) - 1).$$

In this work we will use the Limited Memory BFGS (L-BFGS) algorithm due to its low-memory usage for large number of predictors [7]. Similar to ideas discussed in [17], one can also seek a sparse representation of β by including a $\mathbb{L}_1$ or $\mathbb{L}_2$ penalty on $\mathbf{b}$ in Eqn 5.2.3. For example, if we choose to include an $\mathbb{L}_2$ penalty in Eqn 5.2.3 would be expressed as $\mathbf{b}^* = \arg\max_{\mathbf{b}} \sum_{i=1}^{n} \log\left(\phi\left(y_i \mathbf{b}^\mathsf{T} \mathbf{z}_i\right)\right) - \lambda\|\mathbf{b}\|^2$. The gradient of the log- likelihood would then be $\nabla L(\mathbf{b}) = \sum_{i=1}^{n} -y_i \mathbf{z}_i(\phi(y_i \mathbf{b}^\mathsf{T} \mathbf{z}_i) - 1) - \lambda \mathbf{b}$.

To perform the overall minimization, we alternate between finding the optimal warping functions $\{\gamma_i\}$ and finding the optimal $\mathbf{b}$. The algorithm for computing optimal α, β, and $\{\gamma_i\}$ is given in Algorithm 5.3. As mentioned previously, we have an extra degree of freedom in selecting β as our objective function is invariant to random warpings and we choose the β that corresponds the element of the set where the mean of the warping functions is identity.

Algorithm 5.3 Elastic Functional Logistic Regression

1: Initialization Step: set $\{\gamma_i\} = \gamma_{id}$, calculate SRSFs $\{q_i\}$, set $l = 1$.
2: **while** $\|L^{(l+1)} - L^{(l)}\|^2 < \epsilon$ **do**
3: Find $\alpha^{(l)}$ and $\beta^{(l)}(t)$ using chosen basis and L-BFGS
4: Find $\{\gamma_i\}^{(l)}$ using Dynamic Programming for Eqn. 5.2.2 for each $i = 1, \ldots, n$
5: $l = l + 1$
6: **end while**
7: Find the mean (γ_μ) of $\{\gamma_i^*\}$ using Algorithm 3.1
8: Update $\gamma_i^* \mapsto \gamma_i^* \circ \gamma_\mu^{-1}$
9: Update $\beta^* = (\beta^* \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma_\mu^{-1}}}$
10: Compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma_i^*}}$ and aligned functions $\tilde{f}_i(t) = f_i \circ \gamma_i^*$

This procedure results in four items: α^*, β^*, $\{\gamma_i^*\}$, and $\{\tilde{f}_i\}$.

5.2.2 Prediction

Once the model parameters have been estimated, the model can then be used to predict response variables for new prediction functions. In case there is no phase variation in the predictor function, the class probability can be predicted using $P(y_i = 1|q_i) = \phi(\alpha + \int_0^1 q_i(t)\beta(t)\,dt)$. This probability is then thresholded to determine the class (i.e., $y_i = 1$ if $P(y_i = 1|q_i) \geq 0.5$, and $y_i = -1$ otherwise). In case there is phase variation, the probability is predicted using

$$P(y_i = 1|q_i, \gamma_i) = \phi(\alpha + \int_0^1 (q_i, \gamma_i)(t)\beta(t)\,dt), \tag{5.2.4}$$

but the question remains on what γ_i should be. We propose the following procedure to address this problem: 1) Take the observed SRSF q_i and find the SRSF from the training set which has the smallest $\mathbb{L}^2$ distance $\|q_i - q_{train}\|^2$. 2) Determine the γ_i to be used for prediction as the corresponding time warping for q_{train} in the training sample. 3) Find the probability, using Eqn. 5.2.4 with the corresponding chosen γ_i. 4) Then, threshold the probability to determine the class label.

5.2.3 Experimental Results

In this section, we present some results using both simulated and several real data sets in the context of functional logistic regression.

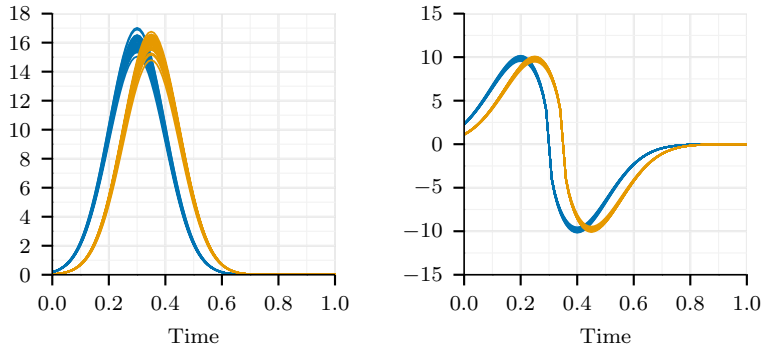
Simulated Data. To illustrate the developed elastic functional regression method we execute Algorithm 5.3 on a simulated two-class data constructed using

$$h_{ij}(t) = a_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu_j)^2}{2\sigma^2}\right), \qquad i = 1, \ldots, 30, \quad j = 1, 2,$$

where the two means are $\mu_1 = 0.3$ and $\mu_2 = 0.35$. The variance $\sigma = 0.1$ is same for both classes, and the coefficients $a_{ij} \sim \mathcal{N}(4, 0.1)$. These functions $\{h_{i1}\}$ and $\{h_{i2}\}$ play the role of predictor functions in our model and the union of the sets is denoted $\{f_i\}$. Specifically, we generate two sets of Gaussian curves with one set having a class label 1 and the other having a class label -1. The generated functions are shown in Fig. 5.6a with corresponding SRSFs in Fig. 5.6b. The blue curves correspond to class 1 and the orange curves correspond to class -1. The functions were then randomly warped to generate the warped $\{f_i\}$, and also SRSFs $\{q_i\}$, and are presented in Figs 5.6c and d, respectively.

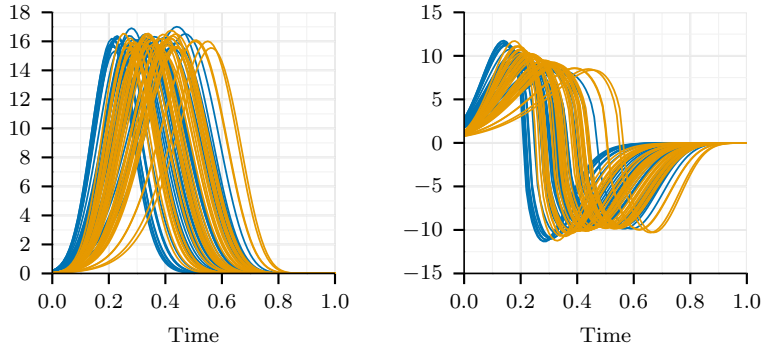
For the estimation of the model, we use a B-spline basis with 20 elements and estimate the parameters using Algorithm 5.3. Fig. 5.7 presents the resulting aligned functions, aligned SRSFs, and corresponding estimated warping functions from Algorithm 5.3 in Panels a, b, and c, respectively. The functions are aligned and clustered into two distinct groups which is similar to how the original data was constructed. Moreover, by examination of Eqn. 5.2.2 we see that the method aligns the samples of class 1 to $\beta(t)$ and those of class -1 to $-\beta(t)$, effectively separating the samples.

To evaluate the performance of the algorithm we performed a 5-fold cross-validation experiment to compare the classification of the elastic method versus the standard functional logistic regression (FLoR) in [17, 23]. Prediction was performed as was described in Section 5.2.2 and since we have



(a) $\{f_i\}$

(b) $\{q_i\}$



(c) Warped $\{f_i\}$

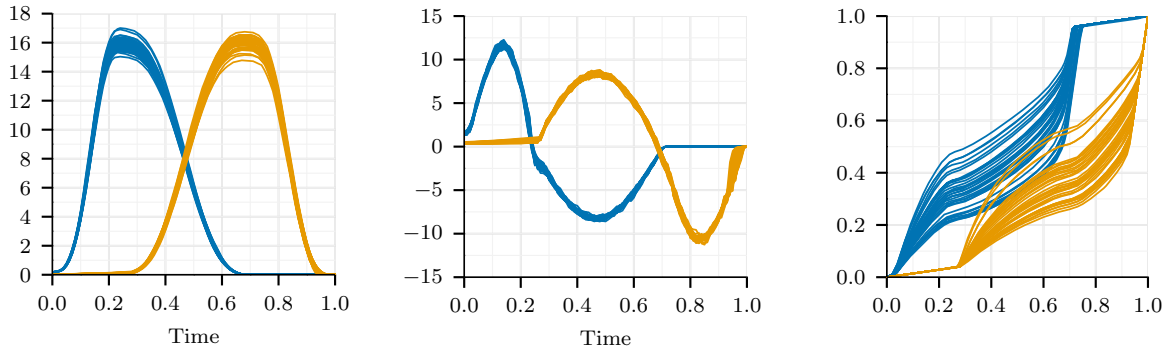
(d) Warped $\{q_i\}$

Figure 5.6: Original Simulated data for logistic regression in (a) $\mathcal{F}$ space and (b) SRSF space with corresponding warped data in (c) $\mathcal{F}$ space and (d) SRSF space.

the labels for the simulated data we can calculate the classification performance. Table 5.2 presents the mean probability of classification across the folds with the corresponding standard deviation in parentheses.

In this table, we compare the elastic FLoR with four methods: 1) standard FLoR on the original data, 2) standard FLoR on the warped data, 3) pre-alignment FLor, and 4) cluster-based alignment FLoR. The alignment-based methods are the same as described in Section 5.1.3, we just substituted standard FLoR for the standard FLR.

For the standard FLoR on the original data, which is the truth in this case, we get the perfect classification performance (100%). For warped data, we see that the elastic method obtains the highest probability of classification compared to the other three methods. Specifically, the pre-alignment method obtains the lowest probability of classification, which is attributed to the



(a) $\{\tilde{f}_i\}$ (b) $\{\tilde{q}_i\}$ (c) $\{\gamma_i\}$

Figure 5.7: Aligned Simulated data in (a) f space and (b) SRSF space with corresponding (c) warping functions resulting from Elastic Functional Logistic Regression.

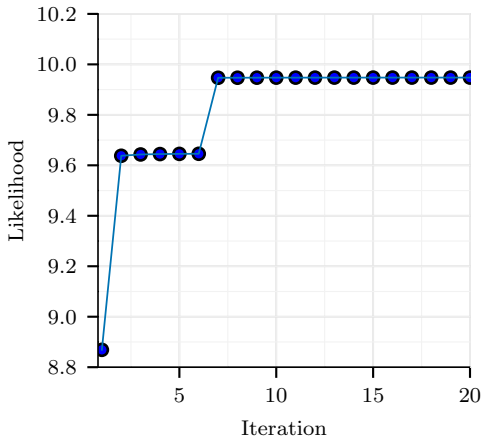


Figure 5.8: Log-Likelihood evolution for Algorithm 5.3 using the simulated data.

pre-alignment destroying the two-class structure of the data. Moreover, it has a higher standard deviation depending on which class the training data was aligned to. The clustering method obtains a higher performance then performing standard FLoR on the un-aligned data. However it demonstrates a larger standard deviation. The higher standard deviation results from some of the functions being assigned to an incorrect cluster. Therefore, the elastic method is able to remove most of the variability of γ and obtain the highest classification performance over current methods.

Real Data. Next, we evaluated the elastic functional logistic regression technique on four sets of real data. The data consists of physiological data specifically, gait and electrocardiogram

Table 5.2: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for simulated data using functional logistic regression.

	Mean (STD)
FLoR Original Data	1.000 (0.000)
FLoR Warped Data	0.613 (0.108)
Pre-Align FLoR	0.425 (0.047)
Cluster FLoR	0.575 (0.047)
Elastic FLoR	**0.950 (0.085)**

(ECG) measurements from various patients. Phase-variability is naturally found in the data, as during collection the signals always start and stop at the different time for each measurement. For example, when measuring a heart beat, one cannot assure that the measurement starts on the same part of the heartbeat for each patient measured. For each of the data sets, we use a B-spline basis with 20 elements for the estimation of the model.

The first data set, Gait, is a collection of gait measurements for patients having Parkinson's disease and those not having Parkinson's disease. It is from the gaitpdb data set on Physionet [20]. This database contains measures of gait from 93 patients with idiopathic Parkison's disease and 73 healthy patients. The gait was measured using vertical ground reaction force records of subjects as they walked at their usual, self-selected pace for approximately 2 minutes on level ground.

The second data set, ECG200, is a collection of ECG measurements of heartbeats for those demonstrating an arrhythmia and those which do not. The data set is from the MIT-BIH Arrhythmia Database available from Physionet. The database contains ECG recordings where each electrocardiogram was recorded from a single patient for a duration of approximately thirty minutes. From the recordings heartbeats were extracted with the most prevalent abnormality-supraventricular premature beat. Additionally, heartbeats were extracted from the recordings that were representative of normal heartbeats. The task is then to distinguish between the abnormality using the heartbeat. Naturally, the heartbeats are not aligned and no alignment was made to the data.

The third data set, TwoLeadECG, is a collection of ECG measurements from the MIT-BIH Long-Term ECG Database available as well from Physionet, which contains long term ECG measurements with beat annotations. Heartbeats were extracted that were annotated normal and abnormal for the two classes.

The fourth data set, ECGFiveDays, is a collection of ECG measurements from a 67 year old male. There are two classes which are simply the data of the ECG measurements which are 5 days apart. The task is then to distinguish between the two days as the wandering baseline was not removed and the heartbeats are not aligned. The dataset is the ECGFiveDays from the UCR Time Series Classification Database [33]. Subsequently, the previous two datasets can also be obtained from the UCR database under the names ECG200 and TwoLeadECG, respectively.

Fig. 5.9 presents the 30 original functions from the Gait, ECG200, TwoLeadECG, and ECG-FiveDays data sets, respectively in the left hand column. The blue curves are the functions from class 1 and the orange curves are the functions from class -1. Using Algorithm 5.3 and the B-spline basis described earlier, the elastic functional logistic regression model was identified. The corresponding warped functions ($f \circ \gamma$) are in the second column with the warping functions (γ) in the third column. For each of the four data sets, the original functions show phase-variability, especially the gait data. After performing the elastic algorithm, the functions are separated, aligned nicely, and show two distinct groups. Indeed, the warping functions also show a distinct grouping for each class.

As was performed previously, we conducted a 5-fold cross-validation experiment to evaluate the probability of classification of the elastic method versus the standard FLoR. Table 5.3 presents the mean probability of classification across the folds with the corresponding standard deviation; in parentheses for standard FLoR on the original data, two alignment-based methods, and our elastic method. The alignment-based methods are the same as those studied on the simulated dataset.

For all four data sets the elastic method outperforms the other three FLoRs. Subsequently, it has a lower standard deviation across the folds indicating that the model generalized well. For the ECGFiveDays it is interesting that the alignment-based methods fail to achieve the same performance as performing the standard FLoR, indicating that a pre-alignment destroys subtle structure in the data. In contrast, the elastic method is able to find that subtle structure to obtain nearly perfect classification rates. In the other three data sets, the classification rates are improved with pre-alignment, however due the two-step nature and the possible destruction of the underlying class structure they never achieve the performance of the elastic method.

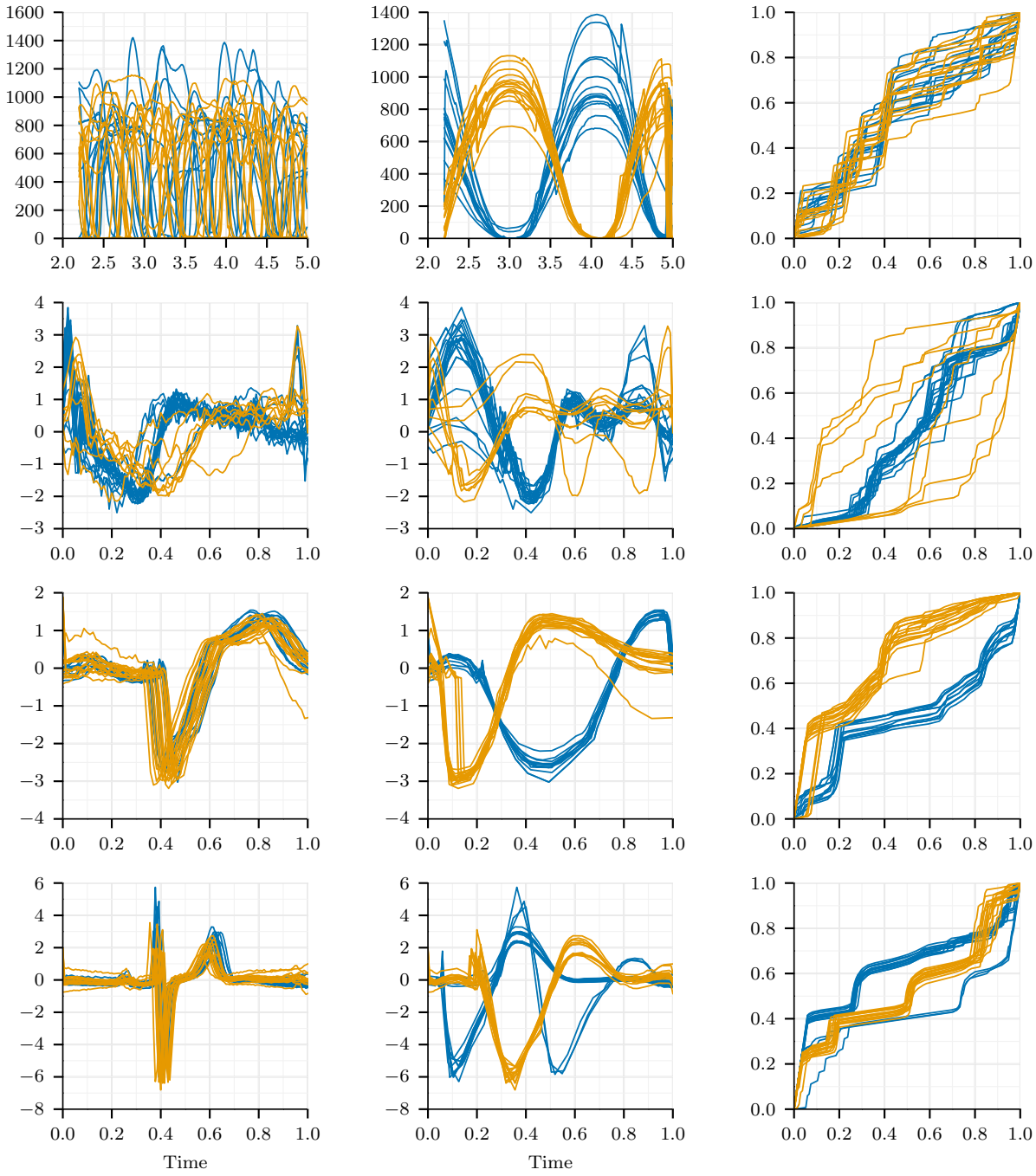


Figure 5.9: Original functions for the Gait, ECG200, TwoLeadECG and ECGFiveDays data sets in the first column (in corresponding descending order) with the corresponding aligned functions and warping functions in the second and third columns, respectively.

Table 5.3: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for the medical data using functional logistic regression.

Data	Method	Mean (STD)
Gait	FLoR Warped Data	0.590 (0.023)
	Pre-Align FLoR	0.689 (0.022)
	Cluster FLoR	0.547 (0.095)
	Elastic FLoR	**0.732 (0.009)**
ECG200	FLoR Warped Data	0.665 (0.075)
	Pre-Align FLoR	0.670 (0.075)
	Cluster FLoR	0.670 (0.081)
	Elastic FLoR	**0.835 (0.046)**
TwoLeadECG	FLoR Warped Data	0.602 (0.021)
	Pre-Align FLoR	0.752 (0.015)
	Cluster FLoR	0.732 (0.040)
	Elastic FLoR	**0.998 (0.003)**
ECGFiveDays	FLoR Warped Data	0.770 (0.093)
	Pre-Align FLoR	0.653 (0.024)
	Cluster FLoR	0.754 (0.098)
	Elastic FLoR	**0.996 (0.002)**

5.3 Elastic Functional Multinomial Logistic Regression

We can extend the elastic functional logistic regression to the case of multinomial response, i.e., y_i has more than two classes. In this case, we have observations $\{(f_i(t),\, y_i)\}$ and the response variable can take on m categories, $y_i \in \{1, \ldots, m\}$, for $i = 1, \ldots, n$. For simplification, we abuse the notation by coding the response variable y as a m-dimensional vector with a 1 in the kth component when $y = k$ and zero, otherwise. Next, lets define the probability of the function f being in class k as

$$P(y^{(k)} = 1 | \{\alpha^{(j)}\}, \{\beta^{(j)}(t)\}, f) = \frac{\exp\left(\alpha^{(k)} + \int_0^1 f(t)\beta^{(k)}(t)\, dt\right)}{1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + \int_0^1 f(t)\beta^{(j)}(t)\, dt\right)}$$

we only need $m - 1$ αs and $\beta(t)$s as we can assume $\alpha^{(m)} = 0$ and $\beta^{(m)}(t) = 0$ without loss of generality.

Using the above probability and the multinomial definition of the problem, we can express the log-likelihood of observations $\{(f_i(t),\, y_i)\}$ as

$$\begin{aligned} L_m(\{\alpha^{(j)}\},\{\beta(t)^{(j)}\}) &= \sum_{i=1}^{n}\left[\sum_{j=1}^{m-1} y_i^{(j)}\left[\alpha^{(j)}+\int_0^1 f_i(t)\beta^{(j)}(t)\,dt\right]\right. \\ &\quad \left. -\log\left(1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)}+\int_0^1 f_i(t)\beta^{(j)}(t)\,dt\right)\right)\right]. \end{aligned}$$

As with logistic regression case, this model assumes that the functions, f_i, are aligned and have no phase-variability. If we incorporate time warping into the model using the SRSF framework as motivated earlier we get the following probability of q (the SRSF of f) being in class k as

$$P(y^{(k)}=1|\{\alpha^{(i)}\},\{\beta(t)^{(i)}\},q,\gamma)=\frac{\exp\left(\alpha^{(k)}+\int_0^1(q,\gamma)(t)\beta^{(k)}(t)\,dt\right)}{1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)}+\int_0^1(q,\gamma)(t)\beta^{(j)}(t)\,dt\right)}.$$

Given observations $\{(f_i(t),\, y_i)\}$ and the associated SRSFs $\{q_i(t)\}$, we can maximize the following log-likelihood to identify the model parameters $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, and $\{\gamma_i^*\}$,

$$\begin{aligned} L &= \sum_{i=1}^{n}\left[\sum_{j=1}^{m-1} y_i^{(j)}\left[\alpha^{(j)}+\int_0^1 (q_i,\gamma_i)(t)\beta^{(j)}(t)\,dt\right]\right. \\ &\quad \left. -\log\left(1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)}+\int_0^1 (q_i,\gamma_i)(t)\beta^{(j)}(t)\,dt\right)\right)\right]. \end{aligned} \tag{5.3.1}$$

5.3.1 Maximum-Likelihood Estimation Procedure

In this multinomial model, we need to estimate $\{\alpha^{*(j)}\},\{\beta^{*(j)}(t)\}$, and $\{\gamma_i^*\}$ using Eqn. 5.3.1. Similar to the binomial case in Section 5.2, we propose an iterative procedure to update $\{\alpha^{*(j)}\},\{\beta^{*(j)}(t)\}$ and $\{\gamma_i^*\}$ alternatively.

Optimization over γ_i. First, we will assume that the set of $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$ are fixed and our goal is to find the set of optimal warping functions, $\{\gamma_i^*\}$. Similar to optimization over γ_i for the elastic functional logistic regression, we can estimate each γ_i individually for each i, since the outer summation in Eqn. 5.3.1 is over i. This optimization can be expressed as

$$\begin{aligned} \gamma_i^* &= \arg\max_{\gamma_i}\sum_{j=1}^{m-1} y_i^{(j)}\left[\alpha^{(j)}+\int_0^1 (q_i,\gamma_i)(t)\beta^{(j)}(t)\,dt\right] \\ &\quad -\log\left(1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)}+\int_0^1 (q_i,\gamma_i)(t)\beta^{(j)}(t)\,dt\right)\right). \end{aligned}$$

We will use the standard gradient ascent method to solve this optimization problem. We will represent an element $\gamma \in \Gamma$ by the square-root of its derivative $\psi = \sqrt{\dot{\gamma}}$. We use this representation as it simplifies the complicated geometry of Γ to a unit sphere as shown in Chapter 3.1.

By taking this transformation the maximization in Eqn. 5.3.2 over ψ_i can be written as

$$H(\psi_i) = \max_{\psi_i} \sum_{j=1}^{m-1} y_i^{(j)} \left[\alpha^{(j)} + A^{(j)}(\psi_i)\right] - \log\left(1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + A^{(j)}(\psi_i)\right)\right), \tag{5.3.2}$$

where $A^{(k)}(\psi_i) = \int_0^1 q_i(\int_0^t \psi_i^2(s)\,ds)\psi(t)\beta^{(k)}(t)\,dt$.

The gradient of Eqn. 5.3.2 is

$$\begin{aligned} h = \frac{\partial H(\psi_i)}{\partial \psi_i} &= \sum_{i=1}^{m-1} y_i^{(j)} \frac{\partial A^{(j)}(\psi_i)}{\partial \psi_i} - \frac{1}{1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + A^{(j)}(\psi_i)\right)} \\ &\quad \sum_{j=1}^{m-1} \left[\exp\left(\alpha^{(j)} + A^{(j)}(\psi_i)\right) \frac{\partial A^{(j)}(\psi_i)}{\partial \psi_i}\right], \end{aligned}$$

where

$$\frac{\partial A^{(k)}(\psi_i)}{\partial \psi_i} = 2\psi_i(t) \int_t^1 q\left(\int_0^x \psi_i^2(s)\,ds\right)\psi_i(x)\beta^{(k)}(x)\,dx + q\left(\int_0^t \psi_i^2(s)\,ds\right)\beta^{(k)}(t)$$

We then find the optimal ψ_i^* and therefore γ_i^* using gradient ascent as described in Algorithm 5.4.

Algorithm 5.4 Optimization over γ_i for Elastic Functional Multinomial Logistic Regression

1: Set $\psi_i^{(0)} = \psi_{id}$ and set $l = 1$
2: **while** $\|H(\psi_i^{(l+1)}) - H(\psi_i^{(l)})\|^2 < \epsilon$ **do**
3: Calculate the gradient h
4: Find the tangent vector in the direction of h at $\psi_i^{(l)}$ using $h - \left\langle h, \psi_i^{(l)} \right\rangle \psi_i^{(l)}$
5: Update ψ_i component according to $\psi_i^{(l+1)} = \cos(\delta\|h\|)\psi_i^{(l)} + \sin(\delta\|h\|)\frac{h}{\|h\|}$ for a step size $\delta > 0$. This update is simply the exponential map on that sphere at the point $\psi_i^{(l)}$ applied to the tangent vector
6: $l = l + 1$
7: **end while**
8: Calculate $\gamma_i^* = \int_0^t \psi_i^l(s)^2\,ds$

Optimization over $\alpha^{(j)}$ and $\beta^{(j)}(t)$. Now, assuming that the $\{\gamma_i\}$ are fixed, we will focus on the solution of finding the optimal $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$. As with the elastic functional logistic regression problem we will use a basis-based approach. Let, $\beta^{(k)}(t) = \sum_{i=1}^{p} b_i^{(k)} \theta_i$, where θ_i is the ith basis function and $b_i^{(k)}$ is the corresponding coefficient. Therefore, we can re-express the maximization problem in Eqn. 5.3.1 as

$$\mathbf{b}^* = \arg\max_{\mathbf{b}} \sum_{i=1}^{n} \left[\sum_{j=1}^{m-1} y_i^{(j)} \mathbf{b}^{(j)\mathsf{T}} \mathbf{z}_i - \log \left(1 + \sum_{j=1}^{m-1} \exp\left(\mathbf{b}^{(j)\mathsf{T}} \mathbf{z}_i \right) \right) \right], \tag{5.3.3}$$

where $\mathbf{b} = [\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m-1)}]^\mathsf{T}$, $\mathbf{b}^{(k)} = [\alpha^{(k)}, b_1^{(k)}, \ldots, b_p^{(k)}]^\mathsf{T}$, and $\mathbf{z}_i = [1, \int (q_i, \gamma_i)(t)\theta_1(t)\, dt, \ldots, \int (q_i, \gamma_i)(t)\theta_p(t)\, dt]^\mathsf{T}$. There is no direct solution to solving this optimization and has to be performed numerically. Since, the function is concave we will use the L-BFGS algorithm to find the solution numerically. To use this algorithm we need the gradient of the log-likelihood. We need to find the partial derivative of the log-likelihood for each $\mathbf{b}^{(k)}$,

$$\frac{\partial L_m(\mathbf{b})}{\partial \mathbf{b}^{(k)}} = \sum_{i=1}^{n} \left[y_i^{(k)} \mathbf{z}_i - \frac{1}{1 + \sum_{j=1}^{m-1} \exp\left(\mathbf{b}^{(j)\mathsf{T}} \mathbf{z}_i \right)} \exp\left(\mathbf{b}^{(k)\mathsf{T}} \mathbf{z}_i \right) \mathbf{z}_i \right].$$

To perform the overall minimization, we alternate from finding the optimal warping functions, $\{\gamma_i\}$, and the optimal $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$. The algorithm for computing the optimal $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, and $\{\gamma_i\}$ is given in Algorithm 5.5. As mentioned previously, we have an extra degree of freedom in selecting each of the $\beta^{(j)}$s as our objective function is invariant to random warpings and we choose each $\beta^{(j)}$ that corresponds the element of the set where the mean of the warping functions is identity.

This procedure results in four items: $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, $\{\gamma_i^*\}$, and $\{\tilde{f}_i\}$.

5.3.2 Prediction

After the model has been identified, one can use it to predict classes of future functional observations. Again, the answer is simple if the test data has not been observed with phase variation as the class probabilities would be computed using

$$P(y_i = k|q_i) = \phi(\alpha^{(k)} + \int_0^1 q_i(t)\beta^{(k)}(t)\, dt), \quad k = 1, \ldots, m, \tag{5.3.4}$$

and the class with maximum probability determines the class (i.e., $k^* = \arg\max_k P(y_i = k|q_i)$). However, a problem arises when the input has been observed with phase variation. The probability

Algorithm 5.5 Elastic Functional Multinomial Logistic Regression

1: Initialization Step: set $\{\gamma_i\} = \gamma_{id}$, calculate SRSFs $\{q_i\}$, set $l = 1$.
2: **while** $\|L_m^{(l+1)} - L_m^{(l)}\|^2 < \epsilon$ **do**
3: Find $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$ using chosen basis and L-BFGS
4: Find $\{\gamma_i\}^{(l)}$ using Algorithm 5.4 for each $i = 1, \ldots, n$
5: $l = l + 1$
6: **end while**
7: Find the mean (γ_μ) of $\{\gamma_i^*\}$ using Algorithm 3.1
8: Update $\gamma_i^* \mapsto \gamma_i^* \circ \gamma_\mu^{-1}$
9: Update each $\beta^{*(j)} = (\beta^{*(j)} \circ \gamma_\mu^{-1})\sqrt{\dot{\gamma_\mu^{-1}}}$
10: Compute the aligned SRSFs using $\tilde{q}_i \mapsto (q_i \circ \gamma_i^*)\sqrt{\dot{\gamma_i^*}}$ and aligned functions using $\tilde{f}_i = f_i \circ \gamma_i^*$.

would be computed using

$$P(y_i = k|q_i, \gamma_i) = \phi(\alpha^{(k)} + \int_0^1 (q_i, \gamma_i)(t)\beta^{(k)}(t)\, dt), \quad k = 1, \ldots, m, \tag{5.3.5}$$

but the question remains on what γ_i should be.

The process is similar to that of elastic functional logistic regression. We propose the following: 1) Take the observed SRSF q_i and find the SRSF from the training set which has the smallest $\mathbb{L}^2$ distance $\|q_i - q_{train}\|^2$. 2) Determine the γ_i to be used for prediction as the corresponding time warping for q_{train} in the training sample. 3) Find the probability for each class, using Eqn. 5.3.5 with the corresponding chosen γ_i. 4) Determine the class membership using $k^* = \arg\max_k P(y_i = k|q_i, \gamma_i)$.

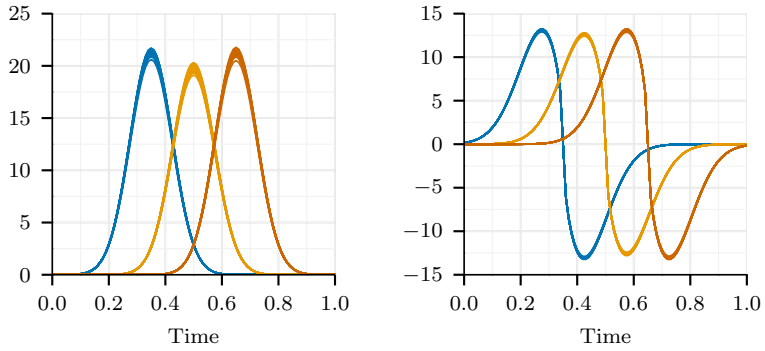
5.3.3 Experimental Results

In this section, we present results on a simulated data set and two real data sets in performing elastic functional multinomial logistic regression.

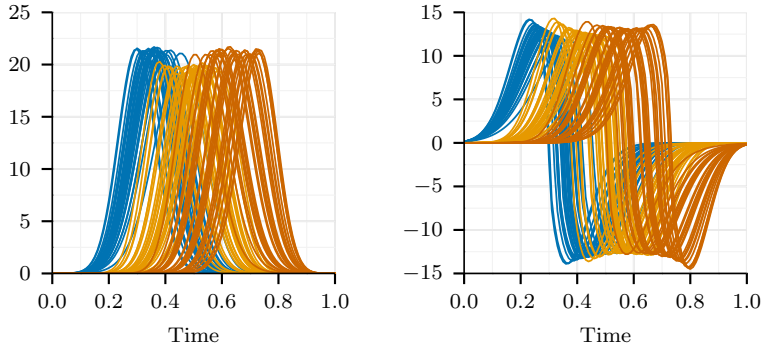
Simulated Data. To illustrate the developed elastic functional regression method we executed Algorithm 5.5 on a simulated three-class data constructed using

$$h_{ij}(t) = a_{ij}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(t-\mu_j)^2}{2\sigma^2}\right), \qquad i = 1, \ldots, 30, \quad j = 1, 2, 3,$$

where the three means $\mu_1 = 0.3, \mu_2 = 0.5$, and $\mu_3 = 0.65$, and the variance $\sigma = 0.075$ is same for the three classes. The coefficients $a_{ij} \sim \mathcal{N}(c_j, 0.1)$ with $c_1 = 4, c_2 = 3.7$, and $c_3 = 4$. To make the



(a) $\{f_i\}$ (b) $\{q_i\}$

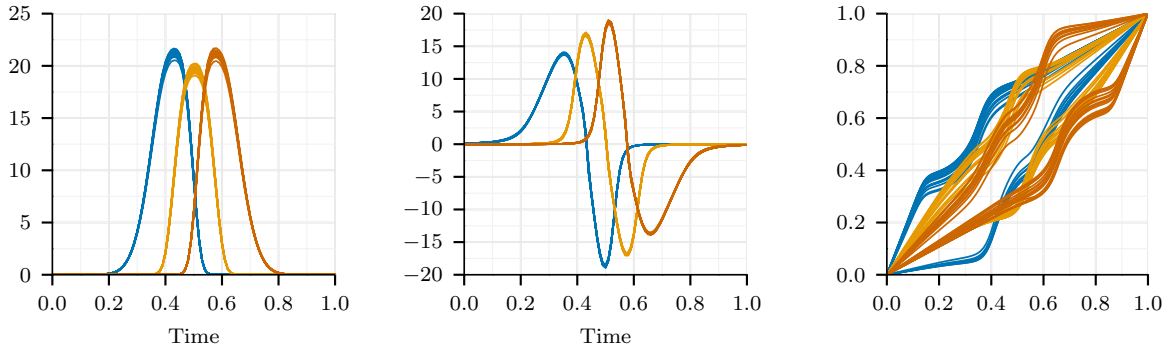


(c) Warped $\{f_i\}$ (d) Warped $\{q_i\}$

Figure 5.10: Original Simulated data for multinomial logistic regression in (a) f space and (b) SRSF space with corresponding warped data in (c) f space and (d) SRSF space.

notation consistent, we use $\{f_i\}$ to denote all observations which is the union of $\{h_{i1}\}$, $\{h_{i2}\}$, and $\{h_{i3}\}$. Specifically, we generated three sets of Gaussian curves with each one having the label 1, 2, or 3. The generated functions are shown in Fig. 5.10a with corresponding SRSFs in Fig. 5.10b. The blue curves are the functions from class 1, the orange curves from class 2, and the red curves from class 3. The functions were the randomly warped to generate the warped $\{f_i\}$ and $\{q_i\}$ and are presented in Figs 5.10c and d, respectively.

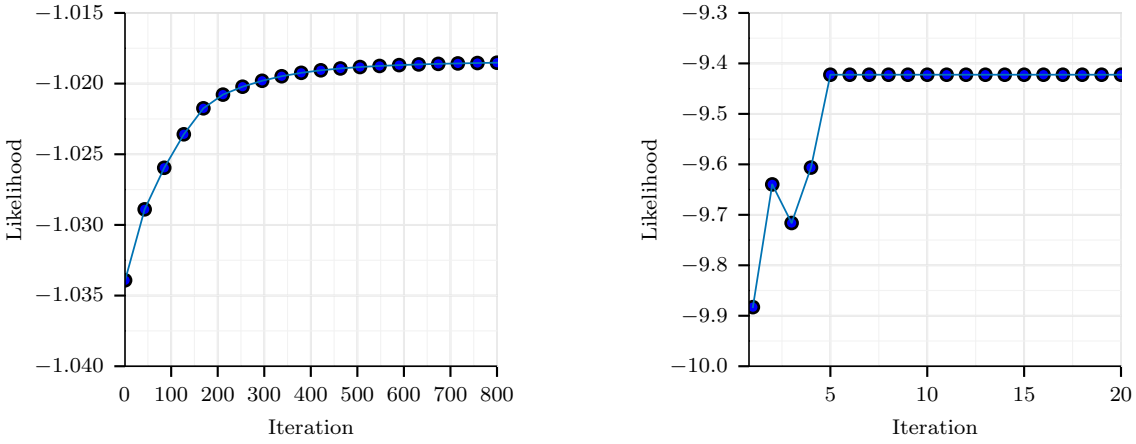
Fig. 5.11 presents the resulting aligned functions, aligned SRSFs, and corresponding estimated warping functions from Algorithm 5.5 in Panels a, b, and c, respectively. A B-spline basis with 20 elements was used for the model identification. The functions are aligned and clustered into three distinct groups which are similar to how the original data was constructed. This is similar to the effect of the elastic functional logistic estimation of γ, just extended to the multi-class case.



(a) Warped $\{f_i\}$ (b) Warped SRSF $\{q_i\}$ (c) Warping Functions $\{\gamma_i\}$

Figure 5.11: Warped Simulated data in (a) f space and (b) SRSF space with corresponding (c) warping functions resulting from Elastic FMLoR.

Fig. 5.12a and b presents the evolution of the log-likelihood for Algorithms 5.4 and 5.5, respectively. The evolution of the log- likelihood for the gradient ascent algorithm for an individual γ_i (Algorithm 5.4) converged in 800 iterations for a step size of 0.01. The evolution for the entire elastic functional multinomial logistic regression algorithm converged in 20 iterations.



(a) (b)

Figure 5.12: Evolution of the likelihood for a) optimization over γ_i (Algorithm 5.4) and b) elastic functional multinomial logistic regression (Algorithm 5.5).

As done previously, we performed a 5-fold cross-validation experiment to evaluate the probability of classification of the elastic method versus the standard functional multinomial logistic

regression (FMLoR). Table 5.4 presents the mean probability of classification across the folds with the corresponding standard deviation in parentheses. The alignment-based methods are the same as described in Section 5.1.3, just substituting standard FMLoR for the standard FLoR.

For the standard FMLoR on the original data, we still get the perfect classification performance (100%). For warped data, we find both pre-alignment methods perform higher versus operating on the given warped data, which shows that some alignment can benefit the classification. Overall, we still see that the elastic method obtains the highest probability of classification compared to the other three methods.

Table 5.4: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for simulated data using functional mulinomial logistic regression.

	Mean (STD)
FMLoR Original Data	1.000 (0.000)
FMLoR Warped Data	0.705 (0.129)
Pre-Align FMLoR	0.814 (0.078)
Cluster FMLoR	0.723 (0.102)
Elastic FMLoR	**0.963 (0.023)**

Real Data. Finally, we evaluated the elastic functional multinomial logistic regression technique on two sets of real data. The data consists of physiologic data similar to those used to test the logistic regression method. As before, for each of the data sets we used a B-spline basis with 20 elements for the estimation of the model.

The first data set is a collection of gait measurements for patients having Parkinson's disease, Amyotrophic lateral sclerosis, Huntington's disease, and healthy controls. The data is from the gaitndd data set on Physionet [20]. This database contains measures of gait from 15, 20, 13, and 16 patients for the respective diseases. The gait was measured using vertical ground reaction force records of subjects as they walked at their usual pace.

The second data set is a collection of ECG measurements from multiple torso-surface sites. There are measurements from 4 different people which are the 4 different classes. The data set is from the 2007 Physionet CinC challenge and is also found as the CinC dataset from the UCR Time Series Classification Database [33].

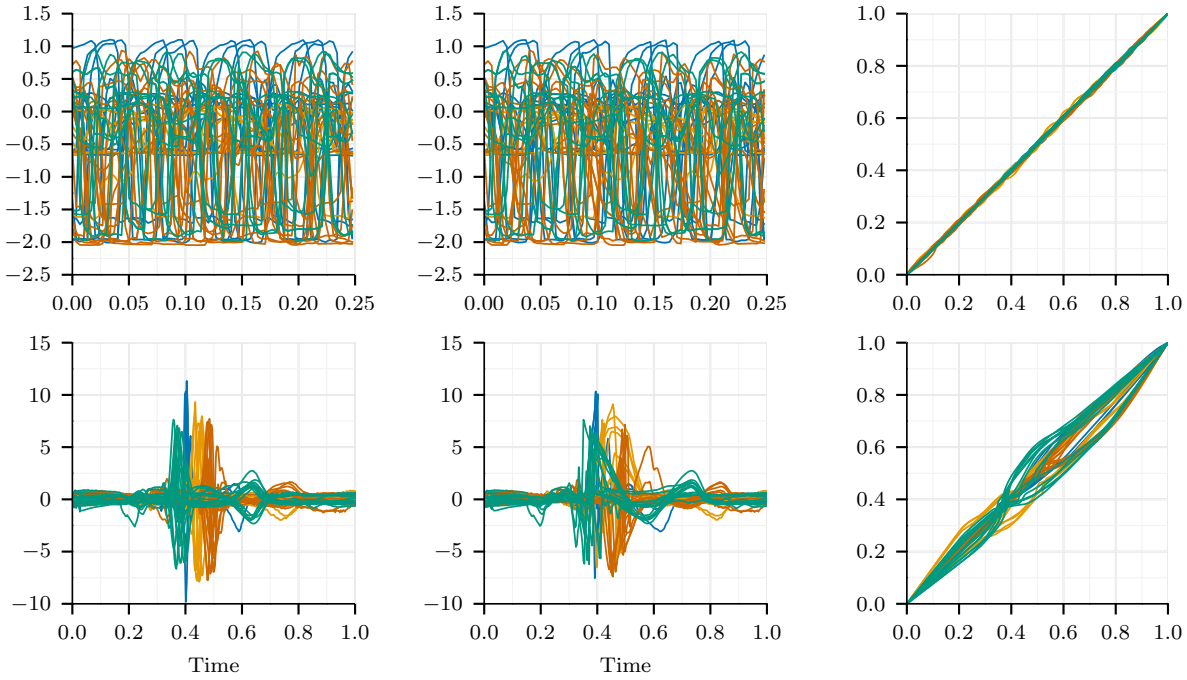


Figure 5.13: Original functions for the Gaitndd and CinC data sets in the first column (in corresponding descending order) with the corresponding aligned functions and warping functions in the second and third columns, respectively.

Fig. 5.13 presents the 48 original functions from the Gaitndd and CinC data sets, respectively in the left hand column. The blue curves are the functions from class 1, the orange curves are the functions from class 2, the red curves are the functions from class 3, and the green curves are the functions from class 4. Using Algorithm 5.5 and the B-spline basis described earlier, the elastic functional multinomial logistic regression model was identified. The corresponding warped functions ($f \circ \gamma$) are in the second column with the warping functions (γ) in the third column. For both of the data sets the original functions show phase-variability, especially the gait data as was seen in the previously studied gait data in Section 5.2.3. After performing the elastic algorithm the the functions there is a little alignment of the gait data due the complicated structure. The CinC data is aligned into four groups and the warping functions exhibit four different clusters.

As before we conducted a 5-fold cross-validation experiment to evaluate the probability of classification of the elastic method versus, the standard functional multinomial logistic regression (FMLoR). Table 5.5 presents the mean probability of classification across the folds with the

corresponding standard deviation in parentheses for standard FMLoR, the two alignment-based methods, and our elastic method.

Table 5.5: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for the medical data using functional multinomial logistic regression.

Data	Method	Mean (STD)
Gaitndd	FMLoR Warped Data	0.395 (0.055)
	Pre-Align FMLoR	0.432 (0.065)
	Cluster FMLoR	0.417 (0.033)
	Elastic FMLoR	**0.478 (0.073)**
CinC	FMLoR Warped Data	0.446 (0.050)
	Pre-Align FMLoR	0.469 (0.069)
	Cluster FMLoR	0.442 (0.058)
	Elastic FMLoR	**0.947 (0.036)**

For both data sets, the elastic method outperforms the standard FMLoR and the alignment based FMLoR. In particular, the classification accuracy is nearly doubled in the CinC data. Moreover, this improvement has a lower standard deviation across the folds indicating that the model generalized well for the data. Overall, the classification rates are improved when the model accounts for the phase-variability in the model.

CHAPTER 6

ELASTIC REGRESSION WITH OPEN CURVES

In this chapter we briefly review a Riemannian framework for elastic shape analysis and modeling of open planar curves. This framework results in metrics, statistics, and models that are invariant to arbitrary rotation, scaling, translation, and re-parametrization (compositional noise) of individual curves. For a more complete review, we refer the reader to [62]. This is an extension of the functional work in Chapter 2 from $\mathbb{R}^1$ to $\mathbb{R}^2$ and provides the same advantage of simultaneous registration and comparisons of the shapes of curves with respect to the elastic metric as the functional data. This has the effect of stretching/compressing and bending parts of curves to match each other in an optimal fashion and is the same as the stretching/compressing of peaks in the functional data for alignment. After the review, we develop elastic regression models using curves as predictors. We then extend the functional linear, logistic and multinomial logistic regression models presented in Chapter 5 to the case when curves are used as predictors instead of functions and demonstrate classification results using shape data.

6.1 Elastic Shape Analysis of Open Curves

Consider an absolutely continuous, parametrized curve $\xi : [0,1] \to \mathbb{R}^2$. Define the set of all re-parameterizations as the set of diffeomorphisms

$$\Gamma = \{\gamma : [0,1] \to [0,1] | \gamma(0) = 0, \gamma(1) = 1, \gamma \text{ increasing}\}. \tag{6.1.1}$$

By definition, γ and γ^{-1} are absolutely continuous. Define a re-parameterization of the curve ξ as the composition $\xi \circ \gamma$, where $\gamma \in \Gamma$. A major problem that arises under this framework is if we want to take the distance between two curves ξ_1 and ξ_2 using standard metrics. Most papers use the standard $\mathbb{L}^2$ metric $\|\xi_1 - \xi_2\|$ to find the distance, however, the standard $\mathbb{L}^2$ metric is not isometric with respect to the group action of Γ. That is, for $\gamma \in \Gamma$, $\|\xi_1 - \xi_2\| \neq \|\xi_1 \circ \gamma - \xi_2 \circ \gamma\|$. Since, we desire to be invariant to re-parameterization and the $\mathbb{L}^2$ metric is not isometric with respect to the group action, it is not possible to compute distances that are invariant to re-parametrization.

In order to achieve this property, Srivastava et. al [62] introduced a novel representation of curves. They represent a shape by its square-root velocity function (SRVF)

$$q(t) = \frac{\dot{\xi}(t)}{\sqrt{|\dot{\xi}(t)|}}. \tag{6.1.2}$$

This is similar to the SRSF defined in Chapter 2.2.2 extended to $\mathbb{R}^n$ with a similar group action given as $(q, \gamma) \equiv (q \circ \gamma)\sqrt{\dot{\gamma}}$. It is easy to show that for two SRVF's q_1 and q_2, $\|q_1 - q_2\| = \|(q_1, \gamma) - (q_2, \gamma)\|$. Therefore the action of the group Γ is isometric on the space of SRVFs.

Another important motivation for the SRVF representation is that that elastic Riemannian metric (a metric that measures a combination of stretching and bending to optimally deform on curve into another) is equal to the standard $\mathbb{L}^2$ metric and therefore much simpler and easier to compute. Since $|q(t)|^2 = |\dot{\xi}(t)|$, the square of the $\mathbb{L}^2$-norm of any SRVF is equal to the length of the corresponding curve ξ. That is, $\|q\|^2 = \int_0^1 |q(t)|^2 \, dt = \int_0^1 |\dot{\xi}(t)| \, dt = L_\xi$. Therefore, the $\mathbb{L}^2$ norm of SRVF's of unit-length curves is one, and the space of such curves is a Hilbert Sphere (unit sphere in infinite dimensional function space)

$$\mathbb{S}^\infty = \{q \in \mathbb{L}^2([0,1], \mathbb{R}^2) | \|q\| = 1\}. \tag{6.1.3}$$

The space $\mathbb{S}^\infty$ represents the SRVF's of all unit-length open curves, and once endowed with the $\mathbb{L}^2$ Riemannian metric, becomes a Riemannian manifold.

A geodesic on a Riemannian manifold is defined as shortest length path with respect to the chosen metric between two points on the manifold. Geodesic distance is the length of the path and we use this as the distance between shapes. Since we know the geometric structure of our manifold, $\mathbb{S}^\infty$, we can easily express geodesic analytically. With respect to the $\mathbb{L}^2$ metric, the geodesic between two points on the unit sphere is defined as the arch segment of the great circle that passes through those two points. Therefore, the geodesic distance is the arc length of this path and mathematically the geodesic path $\nu : [0,1] \to \mathbb{S}^\infty$ with $\nu(0) = q_1$ and $\nu(1) = q_2$ is given by

$$\nu(\tau) = \frac{1}{\sin(\theta)}[\sin((1-\tau)\theta)q_1 + \sin(\tau\theta)q_2] \tag{6.1.4}$$

where $\theta = \cos^{-1}(\langle q_1, q_2 \rangle)$ is the arc length between q_1 and q_2 and $\langle \cdot, \cdot \rangle$ is the $\mathbb{L}^2$ inner product. Therefore, we define the geodesic distance as $d_{\mathbb{S}^\infty} = \theta$. Figure 6.1 shows a diagram of a geodesic path and distance between two points on $\mathbb{S}^\infty$.

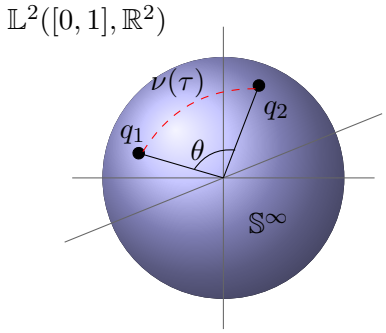


Figure 6.1: A geodesic between two points on $\mathbb{S}^\infty$.

The space $\mathbb{S}^\infty$ is called the *pre-shape space* of elastic curves as the invariants of translation and scaling have been removed. However, the rotation and re-parameterization has yet to be removed. The rotation group $SO(2)$ acts by isometries with respect to the $\mathbb{L}^2$ metric, therefore the quotient space $S_s^O = \mathbb{S}^\infty/(SO(2)\times\Gamma)$ is the similarity- invariant, elastic shape space of open curves. Elements of the quotient space are equivalence classes $[q] = \text{closure}\{O(q,\gamma)|O \in SO(2), \gamma \in \Gamma\}$, where $q \in \mathbb{S}^\infty$. Therefore the distance between orbits $[q_1]$ and $[q_2]$ in the shape space is

$$d_{S_s^O}([q_1],[q_2]) = \inf_{O\in SO(2),\gamma\in\Gamma} d_{\mathbb{S}^\infty}(q_1, O(q_2,\gamma)) \tag{6.1.5}$$

and $d_{\mathbb{S}^\infty}$ is the geodesic distance. The optimization in Eqn. 6.1.5 is performed via Procrustes rigid body alignment for $SO(2)$ and via dynamic programing in the case of Γ. Both of these methods are described in [62] and we refer the reader to that paper for more details of this framework.

6.2 Elastic Linear Regression using Open Curves

Similar to the functional case, assume one now has a set of observations $(\xi_i(t), y_i)$, for $i = 1,\ldots,n$. In this set $\xi_i(t)$ is a parameterized curve predictor and y_i is a scalar real valued response. We can define a linear regression model for this set of observations as

$$y_i = \alpha + \int \langle \xi_i(t), \beta(t)\rangle \, dt + \epsilon_i, \quad i = 1,\ldots,n. \tag{6.2.1}$$

This is similar to the functional linear model defined in Chapter 5.1 where α is the bias and $\beta(t)$ is regression coefficient curve, one main difference is now it utilizes the $\mathbb{L}^2$ inner product for curves. To identify the model parameters we minimize the SSE

$$\{\alpha^*, \beta^*(t)\} = \arg\min_{\alpha,\beta(t)} \sum_{i=1}^{n} |y_i - \alpha - \int \langle \xi_i(t), \beta(t)\rangle \, dt|^2. \tag{6.2.2}$$

We again are presented with the problem, that for a finite n, it is possible to perfectly interpolate the responses if no restrictions are placed on $\beta(t)$ since $\beta(t)$ is infinite dimensional. Another problem with this model is that the variability due to scaling, rotation, and re-parametrization of the curves is not taken into account. Using approaches similar to the functional case we propose a framework that includes these variabilities into the regression model.

Using the SRVF representation of curves and the $\mathbb{L}^2$ metric, which is the Riemannaian metric we can incorporate scaling, rotation, and re-parameterization into the regression model in Eqn. 6.2.1. The motivation for this representation is due to the isometric property of the metric as was demonstrated in Section 6.1. This gives us a predictive variable that is defined as

$$\begin{aligned} y_i^0 &= \alpha + \int \langle q_i^0, \beta \rangle \, dt \\ y_i &= y_i^0 + \epsilon_i, \quad i = 1, \ldots, n. \end{aligned} \tag{6.2.3}$$

The predictive curves are now observed as

$$q_i = O_i(q_i^0, \gamma_i) \tag{6.2.4}$$

and we identify this model by minimizing the SSE

$$\{\alpha^*, \beta^*(t), \{O_i^*\}, \{\gamma_i^*\}\} = \underset{\alpha, \beta(t), \{O_i\}, \{\gamma_i\}}{\arg\min} \sum_{i=1}^{n} |y_i - \alpha - \int \langle O_i(q_i, \gamma_i), \beta \rangle \, dt|^2. \tag{6.2.5}$$

6.2.1 Maximum-Likelihood Estimation Procedure

In the elastic model, we need to estimate $\alpha^*, \beta^*(t)$, $\{O_i^*\}$, and $\{\gamma_i^*\}$ using Eqn. 6.2.5. Note that the optimization is of significant challenge because α and $\beta(t)$ are parameters in the regression and O_i and γ_i is the rotation and re-parameterization in each observation, respectively. As with the elastic functional regression model we propose an iterative procedure to update them alternatively.

Optimization over rotations and warping functions. First, we will assume that α and $\beta(t)$ are given and our goal is to solve for the set of optimal rotations and warping functions, $\{O_i^*\}$ and $\{\gamma_i^*\}$, respectively. Since the summation in Eqn. 6.2.5 is over i, we can estimate the elements of the sets $\{O_i^*\}$ and $\{\gamma_i\}$ individually for each i by solving

$$\{O_i^*, \gamma_i^*\} = \underset{O_i, \gamma_i}{\arg\min} \, |y_i - \alpha - \int \langle O_i(q_i, \gamma_i), \beta \rangle \, dt|^2. \tag{6.2.6}$$

To first the solve the minimization in Eqn. 6.2.6, we can easily get

$$\begin{aligned}\{O_m, \gamma_m\} &= \underset{O_i,\gamma_i}{\arg\min} \int \langle O_i(q_i, \gamma_i), \beta\rangle \, dt \\ \{O_M, \gamma_M\} &= \underset{O_i,\gamma_i}{\arg\max} \int \langle O_i(q_i, \gamma_i), -\beta\rangle \, dt.\end{aligned}$$

As stated previously, the solution to these two optimization problems comes from Procrustes rigid alignment [62] for rotation and the dynamic programming algorithm [3] for the warping functions. Next, let $y_m = \int \langle O_m(q_i, \gamma_m), \beta\rangle \, dt$ and $y_M = \int \langle O_M(q_i, \gamma_M), \beta\rangle \, dt$, we can find the optimal O_i^* and γ_i^* using the following algorithm:

Algorithm 6.1 Optimization over O_i and γ_i for Elastic Linear Regression for Open Curves

if $y_i > \alpha + y_M$ **then**
    $\gamma_i^* = \gamma_M$, $O_i^* = O_M$
end if
if $y_i < \alpha + y_m$ **then**
    $\gamma_i^* = \gamma_m$, $O_i^* = O_m$
end if
if $\alpha + y_m < y_i < \alpha + y_M$ **then**
    Look for the optimal warping function in the following form:

$$\gamma = s\gamma_M + (1-s)\gamma_m, \quad s \in [0,1]$$

    Find the optimal rotation O_i of (q_i, γ_i) to β using Procrustes rigid alignment
    Let $f(\gamma) = y_i - \alpha - \int \langle O_i(q_i, \gamma_i), \beta\rangle \, dt$ and $g(s) = f(s\gamma_M + (1-s)\gamma_m)$
    Then $g(0) = f(\gamma_m) > 0$, and $g(1) = f(\gamma_M) < 0$ and based on the intermediate value theorem, we can use a secant-type method for finding the optimal s^*, such that $g(s^*) = 0$
end if

Optimization over α and β. Now, assuming that $\{O_i\}$ and $\{\gamma_i\}$ are fixed, we will focus on the solution of finding the optimal α^* and $\beta^*(t)$. We will use use the same basis-based approach as the functional case and we assume that we have a set of basis functions, θ_i, $i = 1, \ldots, p$, such as the Fourier basis or a B-spline basis. Let, $\beta(t) = \sum_{i=1}^{p} b_i\theta_i$, where θ_i is the ith basis function and b_i is the corresponding coefficient. With this we can re-express Eqn. 6.2.3 as

$$y_i = \alpha + \Theta^{\mathsf{T}}\mathbf{b} + \epsilon_i, \tag{6.2.7}$$

where the (i,j)th entry in Θ is $\int \langle (q_i, \gamma_i), \theta_j \rangle \, dt$ and $\mathbf{b} = [b_1, \ldots, b_p]^\mathsf{T}$. We then can estimate α and $\mathbf{b}$ using ordinary least squares. Define $Z = [\mathbf{1} \ \Theta]$, where $\mathbf{1}$ is a vector of ones and $\mathbf{y} = [y_1, \ldots, y_n]^\mathsf{T}$, then the solution for α^* and $\mathbf{b}^*$ is

$$[\alpha^*, \mathbf{b}^*]^\mathsf{T} = (Z^\mathsf{T} Z)^{-1} Z^\mathsf{T} \mathbf{y} \tag{6.2.8}$$

and the optimal $\beta^*(t) = \sum_{i=1}^{p} b_i^* \theta_i$.

6.2.2 Prediction

After the model has been identified, the next question that arises is how is prediction performed. The answer is simple if the test curve has not been observed with rotation and parameterization variability as prediction would be computed using

$$y_i = \alpha + \int_0^1 \langle q_i, \beta \rangle \, dt. \tag{6.2.9}$$

However, a problem arises when the input has been observed with rotation and parametrization variation. The prediction would be performed using

$$y_i = \alpha + \int_0^1 \langle O_i(q_i, \gamma_i), \beta \rangle \, dt, \tag{6.2.10}$$

but the question remains on what γ_i and O_i should be.

To solve this problem, we propose the following: 1) Take the observed SRVF, q_i and find the SRVF from the training set which has the smallest distance using $\|q_i - q_{train}\|^2$. 2) Determine the γ_i and O_i to be used for prediction as the corresponding γ_i and O_i for the closest training sample. 3) Find the predictive value, y_i, using Eqn. 6.2.10 with the corresponding chosen γ_i and O_i.

6.3 Elastic Logistic Regression using Open Curves

Now that we have the linear model defined, we would like to develop a model where the response variable is binary, $y_i \in \{-1, 1\}$, for $i = 1, \ldots, n$. In this case, we want to classify the curves to a specific class given their curve predictor. Following the same line of formulation as the Elastic Functional Logistic Regression model in Chapter 5.2, we will develop the model. As was motivated in the previous section we will use the SRVF representation to include rotation and

re-parameterization. Therefore, we can define the probability of q_i (the SRVF of ξ_i) being in class 1 as

$$P(y=1|q_i, O_i, \gamma_i) = \frac{1}{1+\exp\left(-\left[\alpha + \int_0^1 \langle O_i(q_i,\gamma_i),\beta\rangle\ dt\right]\right)}.$$

This is nothing but the logistic link function $\phi(t) = 1/(1+\exp(-t))$ applied to the conditional mean in a linear regression model: $\alpha + \int_0^1 \langle O_i(q_i,\gamma_i),\beta\rangle\ dt$. Using this relation, and the fact that $P(y=-1|f_i) = 1 - P(y=1|f_i)$, we can express the data likelihood as:

$$\pi(\{y_i\}|\{q_i\},\alpha,\beta,\{O_i\},\{\gamma_i\}) = \prod_{i=1}^{n} \frac{1}{1+\exp\left(-y_i\left[\alpha + \int_0^1 \langle O_i(q_i,\gamma_i),\beta\rangle\ dt\right]\right)}.$$

Assuming we observe a sequence of i.i.d. pairs $\{q_i, y_i\}, i = 1, \cdots, n$, the model is identified by maximizing the log-likelihood according to,

$$\{\alpha^*, \beta^*(t), \{\gamma_i^*\}, \{O_i^*\}\} = \arg\max_{\alpha,\beta(t)} \sum_{i=1}^{n} \left(\arg\max_{O_i,\gamma_i} \log\left(\phi\left(y_i\left[\alpha + \int_0^1 \langle O_i(q_i,\gamma_i),\beta\rangle\ dt\right]\right)\right)\right). \tag{6.3.1}$$

6.3.1 Maximum-Likelihood Estimation Procedure

In the elastic model, we need to estimate α^*, β^*, $\{\gamma_i^*\}$, and $\{O_i^*\}$ using Eqn. 6.3.1. Note that the optimization is of significant challenge because α and $\beta(t)$ are parameters in the regression and γ_i and O_i are the re-parameterization and rotation in each observation, respectively. Again, we propose an iterative procedure to update them alternatively.

Optimization over warping functions and rotations. First, we will assume that α and β are fixed and our goal is to find the set of optimal warping functions and rotations, $\{\gamma_i^*\}$ and $\{O_i^*\}$. We can estimate each γ_i and O_i separately by solving

$$\{O_i^*, \gamma_i^*\} = \arg\max_{O_i,\gamma_i} \left(y_i \left[\int_0^1 \langle O_i(q_i,\gamma_i),\beta\rangle\ dt \right] \right).$$

Note that since $\|O_i(q_i,\gamma_i)\| = \|q_i\|$ ($\|\cdot\|$ denotes the $\mathbb{L}^2$ norm), we can find the optimal O_i and γ_i using

$$\{O_i^*, \gamma_i^*\} = \arg\min_{O_i,\gamma_i} \|O_i(q_i,\gamma_i) - y_i\beta\|^2. \tag{6.3.2}$$

That is, O_i and γ_i^* are the optimal re-parameterization and rotation of the SRVF q_i to match the curve $y_i\beta$. The solution can be effectively computed using a dynamic programming algorithm on a finite grid [3] for γ_i and Procrustes rigid alignment for O_i.

Optimization over α and β. In this step we assume that the $\{\gamma_i\}$ and $\{O_i\}$ are fixed, and adopt a conventional basis-based approach for estimating α^* and β^* as in the previous cases. Let, $\beta(t) = \sum_{i=1}^{p} b_i \theta_i$, where θ_i is the ith basis function and b_i is the corresponding coefficient. Finding the optimal parameter vector is similar to the functional case in Chapter 5.2 and is given as follows:

$$\mathbf{b}^* = \underset{\mathbf{b} \in \mathbb{R}^{p+1}}{\arg\max} \sum_{i=1}^{n} \log \left(\phi \left(y_i \mathbf{b}^\mathsf{T} \mathbf{z}_i \right) \right), \tag{6.3.3}$$

where $\mathbf{b} = [\alpha, b_1, \ldots, b_p]^\mathsf{T}$ and $\mathbf{z}_i = [1, \int_0^1 \langle O_i(q_i, \gamma_i), \theta_1 \rangle \, dt, \ldots, \int_0^1 \langle O_i(q_i, \gamma_i), \theta_p \rangle \, dt]^\mathsf{T}$. Again, there is no analytical solution to this optimization problem and since it is concave we use the L-BFGS algorithm to solve.

To perform the overall minimization we alternate between finding the optimal warping functions and rotations $\{\gamma_i\}$ and $\{O_i\}$ and finding the optimal $\mathbf{b}$. The algorithm for computing optimal α, β, $\{\gamma_i\}$, and $\{O_i\}$ is given in Algorithm 6.2.

Algorithm 6.2 Elastic Curve Logistic Regression

1: Initialization Step: set $\{\gamma_i\} = \gamma_{id}$, calculate SRVFs $\{q_i\}$, set $l = 1$.
2: **while** $\|L_o^{(l+1)} - L_o^{(l)}\|^2 < \epsilon$, where L_o is the log-likelihood **do**
3: Find $\alpha^{(l)}$ and $\beta^{(l)}(t)$ using chosen basis and L-BFGS
4: Find $\{\gamma_i\}^{(l)}$ and $\{O_i\}^{(l)}$ using Dynamic Programming and Procrustes rigid alignment for Eqn. 6.3.2 for each $i = 1, \ldots, n$
5: $l = l + 1$
6: **end while**
7: Compute the aligned SRVFs using $\tilde{q}_i \mapsto O_i^*(q_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$ and aligned curves $\tilde{\xi}_i(t) = O_i^*(\xi_i \circ \gamma_i^*)$

This procedure results in five items: α^*, β^*, $\{\gamma_i^*\}$, $\{O_i^*\}$, and $\{\tilde{\xi}_i\}$.

6.3.2 Prediction

Once the model parameters have been estimated, the model can then be used to predict response variables for new prediction curves. In case there is no parameterization and rotation variation in the predictor function, the class probability can be predicted using $P(y_i = 1|q_i) = \phi(\alpha + \int_0^1 \langle q_i, \beta \rangle \, dt)$. This probability is then thresholded to determine the class (i.e., $y_i = 1$ if $P(y_i = 1|q_i) \geq 0.5$, and $y_i = -1$ otherwise). In case there is rotation and parameterization variability, the probability is predicted using

$$P(y_i = 1|q_i, \gamma_i, O_i) = \phi(\alpha + \int_0^1 \langle O_i(q_i, \gamma_i), \beta \rangle \, dt), \tag{6.3.4}$$

but the question remains on what γ_i and O_i should be. We follow the same pattern as in the previous elastic models where we find the training SRVF nearest the observed SRVF q_i and use the corresponding γ_i and O_i. We then find the probability using Eqn. 6.3.4 using the chosen γ_i and O_i and threshold to determine the class label.

6.3.3 Experimental Results

In this section, we classify real world curve data taken from the MPEG-7 database [26]. The full database has 1300 shape samples, 20 shapes for each class, and 65 shape classes. For each experiment we chose two shape classes from the database to test our logistic regression model. An example of some of the shapes from the MPEG-7 bases is presented in Figure 6.2. Note the differences in rotations and scales of each of the shapes which complicates the analysis.

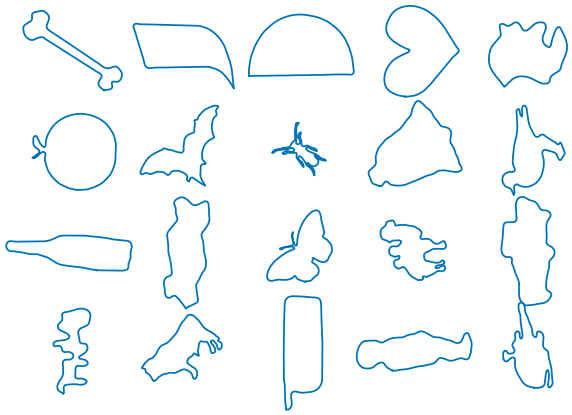
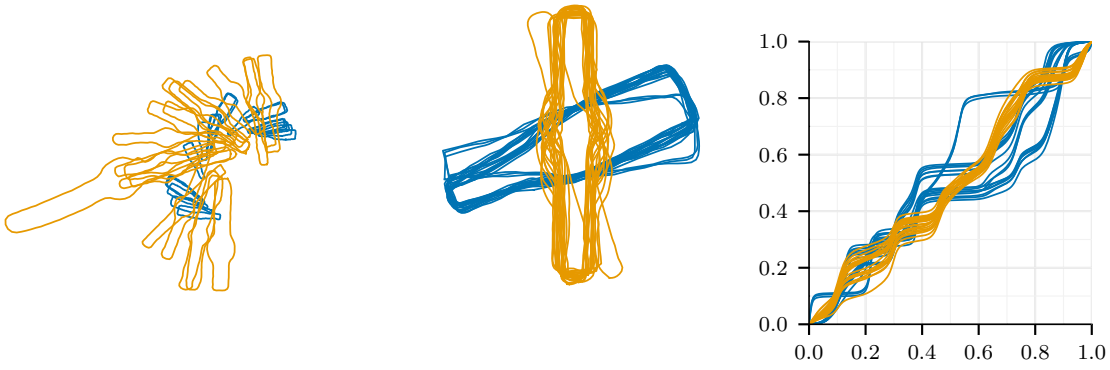


Figure 6.2: Example shapes from the MPEG-7 shape database.

To increase the complexity of the database, we rotated and re-parameterized each of the shape samples randomly. First, we tested the model on distinguishing between a bottle and watch. This is somewhat of a difficult problem considering how similar the two shapes can be. Fig. 6.3a presents the original curves which demonstrate the differences in scale and rotation between the two shapes. For the estimation of the model, we used a B-spline basis with 60 elements and estimated the parameters of the model using Algorithm 6.2. Fig. 6.3b presents the aligned curves in which we notice the curves have been aligned into two distinct groups, which are separated by a rotation. Fig. 6.3c presents the corresponding warping functions and exhibit that the method registers the samples into two classes. Specifically, the method, as in the functional case, registers the samples

of class 1 (bottles) to β and those of class -1 (watches) to $-\beta$, effectively separating the samples. Moreover, the curves are scaled to length 1 in the SRVF transformation which removes the effects of scale in the analysis.



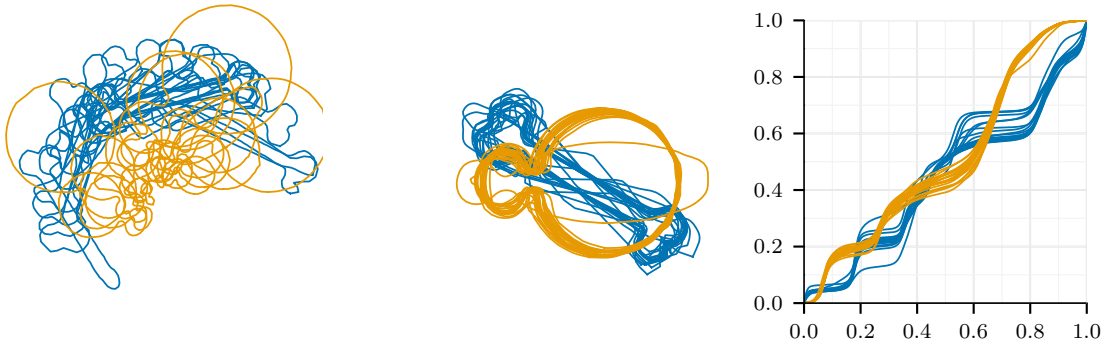
(a) $\{\xi_i\}$ (b) $\{\tilde{\xi}_i\}$ (c) $\{\gamma_i\}$

Figure 6.3: Bottle and wrist-watch curves from the MPEG-7 database, (a) un-registered curves, (b) registered curves, and (c) warping functions resulting from Elastic Curve Logistic Regression.

We conducted a similar test on the bottle and pocket-watch curves from the database. The original curves, registered curves, and warping functions are presented in Panels a, b, and c of Fig. 6.4, respectively. As with the previous set of curves, the aligned curves are aligned and rotated into two distinct groups.

To evaluate the performance of the algorithm, we performed a 5-fold cross- validation experiment to compare the classification of the elastic method versus the standard curve logistic regression (CLoR). By standard we mean calculating the class probability using $\phi(\alpha + \int \langle \xi_i, \beta \rangle \, dt)$ and identifying the model parameters α and β by solving $\arg\max_{\alpha,\beta} \sum_{i=1}^{n} \log(\phi(y_i[\alpha + \int \langle \xi_i, \beta \rangle \, dt]))$ using L-BFGS and a basis representation. Prediction was performed as was described above and since we have the labels for the curves can calculate the classification performance. Table 6.1 presents the mean probability of classification across the folds with the corresponding standard deviation in parentheses for both pairs of shapes studied in Fig. 6.3 and Fig. 6.4 in Panels a and b, respectively.

In this table, we compare the elastic CLoR with three regression methods: 1) standard CLoR on the original data, 2) pre-align CLoR, which pre-aligns the training curves using the method described in [62] and performs standard CLoR. Prediction is performed by taking the warped



(a) $\{\xi_i\}$ (b) $\{\tilde{\xi}_i\}$ (c) $\{\gamma_i\}$

Figure 6.4: Bone and pocket-watch curves from the MPEG-7 database, (a) un-registered curves, (b) registered curves, and (c) warping functions resulting from Elastic Curve Logistic Regression.

sample and aligning it to the mean function found using the alignment and calculating the class using the identified model. Lastly, 4) cluster CLoR, which clusters the training response data, $\{y_i\}$, using k-means, and then aligns the training curves inside each cluster using the same alignment algorithm as the second method. Prediction is performed by taking the sample curve and finding the cluster it is closest to in the training data, then aligning it to the mean curve of the corresponding cluster, and then calculate the class.

We also compared the methods against two metric-based nearest-neighbor classifier (1-NN). The first metric is the elastic distance between two SRVFs from [62] which handles rotation, re-parameterization, translation, and scaling. The second metric is Kendall's shape distance from [32] which is based on landmarks and handles rotation, translation, and scale. Both of these methods require more computation than the regression-based methods as for each test sample the distance must be computed between each curve in the training set. The Elastic distance outperforms the Elastic CLoR in all cases, this is to be expected the regression models are linear and limited in performance. However, our regression model is computationally faster than the elastic metric and our performance can be improved by constructing a non-linear model. Additionally, we outperform Kendall's metric which does not handle the re-parameterization variability and is worse than even the standard regression models.

Table 6.1: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for MPEG-7 data using curve logistic regression.

	Mean (STD)
CLoR Warped Data	0.800 (0.150)
Pre-Align CLoR	0.450 (0.322)
Cluster CLoR	0.600 (0.146)
Elastic CLoR	**0.875 (0.079)**
Elastic 1-NN	1.000 (0.000)
Kendall 1-NN	0.700 (0.0612)

(a) Bottle versus Watch

	Mean (STD)
CLoR Warped Data	0.875 (0.112)
Pre-Align CLoR	0.725 (0.184)
Cluster CLoR	0.775 (0.122)
Elastic CLoR	**1.000 (0.000)**
Elastic 1-NN	1.000 (0.000)
Kendall 1-NN	0.540 (0.0612)

(b) Bone versus Pocket-Watch

For both pairs of data, the standard CLoR on the original data performs well. The alignment-based methods perform worse than the standard method on the original data. Specifically the pre-alignment method obtains the lowest probability of classification, which is attributed to the pre-alignment destroying the two-class structure of the data. Moreover, it has a higher standard deviation depending which class the training data was aligned to. The clustering method obtains a higher performance than performing standard CLoR on the un-aligned data, its variability results from some of the curves being assigned to an incorrect cluster. The elastic method is able to remove most of the variability of γ and O and obtain the highest classification performance over current methods.

6.4 Elastic Multinomial Logistic Regression using Open Curves

We can extend the elastic curve logistic regression to the case where the response takes on more than two classes, i.e., multinomial. In this case, we have observations $\{(\xi_i(t), y_i)\}$ and the response variable can take on m categories, $y_i \in \{1, \ldots, m\}$, for $i = 1, \ldots, n$. As in the functional case, we code the response variable into a m-dimensional vector for simplification where the vector contains a 1 in the kth position if $y_i = k$ and 0 otherwise. Using the SRVF formulation we can extend the

logistic regression model and define the probability of q_i (the SRVF of ξ_i) being in class 1 as

$$P(y^{(k)} = 1|\{\alpha^{(i)}\}, \{\beta(t)^{(i)}\}, q, O, \gamma) = \frac{\exp\left(\alpha^{(k)} + \int_0^1 \left\langle O(q,\gamma), \beta^{(k)} \right\rangle \, dt\right)}{1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + \int_0^1 \left\langle O(q,\gamma), \beta^{(j)} \right\rangle \, dt\right)}.$$

We only need $m-1$ αs and $\beta(t)$s as we can assume $\alpha^{(m)} = 0$ and $\beta^{(m)}(t) = 0$ without loss of generality. Given observations $\{(\xi_i,\ y_i)\}$ and the associated SRVFs $\{q_i\}$, we can maximize the following log-likelihood to identify the model parameters $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, $\{O_i^*\}$, and $\{\gamma_i^*\}$,

$$\begin{aligned} L_{om} \quad = \quad & \sum_{i=1}^{n} \left[\sum_{j=1}^{m-1} y_i^{(j)} \left[\alpha^{(j)} + \int_0^1 \left\langle O_i(q_i,\gamma_i), \beta^{(j)} \right\rangle \, dt \right] \right. \\ & \left. - \log\left(1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + \int_0^1 \left\langle O_i(q_i,\gamma_i), \beta^{(j)} \right\rangle \, dt\right)\right) \right]. \end{aligned} \tag{6.4.1}$$

6.4.1 Maximum-Likelihood Estimation Procedure

In this multinomial model, we need to estimate $\{\alpha^{*(j)}\}, \{\beta^{*(j)}(t)\}$, $\{O_i\}$ and $\{\gamma_i^*\}$ using Eqn. 6.4.1. Similar to the binomial case in Section 5.2, we propose an iterative procedure to update $\{\alpha^{*(j)}\}, \{\beta^{*(j)}(t)\}$, $\{O_i^*\}$, and $\{\gamma_i^*\}$ alternatively.

Optimization over γ_i and O_i. First, we will assume that the set of $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$ are fixed and our goal is to find the set of optimal warping functions, $\{\gamma_i^*\}$ and rotations, $\{O_i^*\}$. Similar to optimization over γ_i and O_i for the elastic curve logistic regression, we can estimate each γ_i and O_i individually for each i, since the outer summation in Eqn. 6.4.1 is over i. This optimization can be expressed as

$$\begin{aligned} \{O_i^*, \gamma_i^*\} \quad = \quad & \underset{O_i,\gamma_i}{\arg\max} \sum_{j=1}^{m-1} y_i^{(j)} \left[\alpha^{(j)} + \int_0^1 \left\langle O_i(q_i,\gamma_i), \beta^{(j)} \right\rangle \, dt \right] \\ & - \log\left(1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + \int_0^1 \left\langle O_i(q_i,\gamma_i), \beta^{(j)} \right\rangle \, dt\right)\right). \end{aligned}$$

We will use the standard gradient ascent method to solve this optimization problem. Again, we will represent an element $\gamma \in \Gamma$ by the square-root of its derivative $\psi = \sqrt{\dot{\gamma}}$, since it simplifies the complicated geometry of Γ to a sphere. This motivation behind this representation is explained in more detail in Chapter 3.1. As a reminder, the set of all ψ_i, with identity element $\psi_{id} = 1$, is a Hilbert Sphere, $\mathbb{S}_\infty$.

By taking this transformation the maximization in Eqn. 6.4.2 over ψ_i and O_i can be written as

$$H(O_i,\psi_i) = \max_{O_i,\psi_i} \sum_{j=1}^{m-1} y_i^{(j)} \left[\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right] - \log\left(1 + \sum_{j=1}^{m-1} \exp\left(\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right)\right), \quad (6.4.2)$$

where $A^{(k)}(O_i,\psi_i) = \int_0^1 \left\langle O_i(q_i(\int_0^t \psi_i^2(s)\,ds)\psi), \beta^{(k)} \right\rangle dt$.

We need to find the partial derivatives of Eqn. 6.4.2 with respect to O_i and ψ_i. For O_i the rotation matrix is defined to be

$$O_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

and the derivative of O_i with respect to θ_i is

$$\frac{\partial O_i}{\partial \theta_i} = \begin{bmatrix} -\sin\theta_i & -\cos\theta_i \\ \cos\theta_i & -\sin\theta_i \end{bmatrix}.$$

Therefore, the partial derivatives of Eqn. 6.4.2 with respect to θ_i and ψ_i are

$$\begin{aligned} \frac{\partial H(O_i,\psi_i)}{\partial \theta_i} &= \sum_{i=1}^{m-1} y_i^{(j)} \frac{\partial A^{(j)}(O_i,\psi_i)}{\partial \theta_i} - \frac{1}{1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right)} \\ &\quad \sum_{j=1}^{m-1}\left[\exp\left(\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right)\frac{\partial A^{(j)}(O_i,\psi_i)}{\partial \theta_i}\right] \quad (6.4.3) \\ \frac{\partial H(O_i,\psi_i)}{\partial \psi_i} &= \sum_{i=1}^{m-1} y_i^{(j)} \frac{\partial A^{(j)}(O_i,\psi_i)}{\partial \psi_i} - \frac{1}{1+\sum_{j=1}^{m-1}\exp\left(\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right)} \\ &\quad \sum_{j=1}^{m-1}\left[\exp\left(\alpha^{(j)} + A^{(j)}(O_i,\psi_i)\right)\frac{\partial A^{(j)}(O_i,\psi_i)}{\partial \psi_i}\right], \quad (6.4.4) \end{aligned}$$

where

$$\frac{\partial A^{(k)}(O_i,\psi_i)}{\partial \theta_i} = \int_0^1 \left\langle \frac{\partial O_i}{\partial \theta_i}\left(q_i\left(\int_0^t \psi_i^2(s)\,ds\right)\psi\right), \beta^{(k)} \right\rangle dt.$$

To find $\frac{\partial A^{(k)}(O_i,\psi_i)}{\partial \psi_i}$ we use the process described in [62]. First, consider the sequence of maps $\psi \overset{\int_0^t \psi^2 ds}{\mapsto} \gamma \overset{\phi}{\mapsto} r$, where $r \equiv \phi(\gamma) = (q\circ\gamma)\sqrt{\dot\gamma}$. For the constant function $\mathbf{1} \in \Psi$ and a tangent vector $u \in T_{\mathbf{1}}(\mathbb{S}_\infty)$ the differential of the first mapping at $\mathbf{1}$ is $2\bar u(t) = 2\int_0^t u(s)ds$. For a tangent vector $w \in T_{\gamma_{id}}(\Gamma)$, the differential of the second mapping at $\gamma_{id} = t$ is $\frac{\partial \tilde q}{\partial t}w + \frac{1}{2}\tilde q \dot w$, where $\tilde q = O(q(\int_0^t \psi(s)^2 ds)\psi$. If we concatenate these two linear maps we obtain the directional partial derivative of $A^{(k)}(O_i,\psi_i)$ in a direction $u \in T_{\mathbf{1}}(\mathbb{S}_\infty)$ as

$$\nabla_\psi A^{(k)}(u) = \int_0^1 \left\langle 2\frac{\partial \tilde q_i}{\partial t}\bar u(t) + \tilde q_i u(t), \beta^{(k)}(t) \right\rangle dt.$$

Since $T_{\mathbf{1}}(\mathbb{S}_\infty)$ is an infinite-dimensional space, we can approximate the directional partial derivative by considering a finite-dimensional subspace of $T_{\mathbf{1}}(\mathbb{S}_\infty)$. Lets form a subspace of $T_{\mathbf{1}}(\mathbb{S}_\infty)$ using $\{(\frac{1}{\sqrt{\pi}}\sin(2\pi nt), \frac{1}{\sqrt{\pi}}\cos(2\pi nt))|n = 1, 2, \ldots, p/2\}$. We then can approximate the derivative using

$$c = \sum_{i=1}^{p} \nabla_\psi A^{(j)}(c_i)c_i,$$

where c_i's are the basis elements of the subspace. Using c in Eqn 6.4.4 for $\frac{\partial A^{(k)}(O_i,\psi_i)}{\partial \psi_i}$ we can then compute the full derivative.

We then find the optimal θ_i^* and ψ_i^* and therefore γ_i^* and O_i^* using gradient ascent as described in Algorithm 6.3.

Algorithm 6.3 Optimization over O_i and γ_i for Elastic Curve Multinomial Logistic Regression

1: Set $\psi_i^{(0)} = \psi_{id}$, $\theta_i = 0$ and set $l = 1$
2: **while** $\|H(O_i^{(l+1)}, \psi_i^{(l+1)}) - H(O_i^{(l)}\psi_i^{(l)})\|^2 < \epsilon$ **do**
3: Calculate $\frac{\partial H(O_i,\psi_i)}{\partial \theta_i}$ and $\frac{\partial H(O_i,\psi_i)}{\partial \psi_i}$ using Eqn. 6.4.3 and Eqn. 6.4.4, respectively
4: Update θ_i according to $\theta_i^{(l+1)} = \theta_i^{(l)} + \delta_O \frac{\partial H(O_i,\psi_i)}{\partial \theta_i}$ for a step size $\delta_O > 0$ and in turn form $O_i^{(l+1)}$
5: Update ψ_i component according to $\psi_i^{(l+1)} = \cos(\delta_g\|h\|)\mathbf{1} + \sin(\delta_g\|h\|)\frac{h}{\|h\|}$ for a step size $\delta_g > 0$ and $h = \frac{\partial H(O_i,\psi_i)}{\partial \psi_i}$. This update is simply the exponential map on that sphere at the point $\mathbf{1}$ applied to the tangent vector
6: $l = l + 1$
7: **end while**
8: Calculate $\gamma_i^* = \int_0^t \psi_i^l(s)^2\, ds$ and $O_i^* = O_i^{(l)}$

Optimization over $\alpha^{(j)}$ and $\beta^{(j)}(t)$. Now, assuming that the $\{\gamma_i\}$ and $\{O_i\}$ are fixed, we will focus on the solution of finding the optimal $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$. As with the elastic curve logistic regression problem, we will use a basis-based approach where we represent each $\beta^{(j)} = \sum_{i=1}^{p} b_i^j \theta_i$. Finding the optimal parameter vector is similar to the functional case and is given as follows:

$$\mathbf{b}^* = \arg\max_{\mathbf{b}} \sum_{i=1}^{n} \left[\sum_{j=1}^{m-1} y_i^{(j)} \mathbf{b}^{(j)\mathsf{T}} \mathbf{z}_i - \log\left(1 + \sum_{j=1}^{m-1} \exp\left(\mathbf{b}^{(j)\mathsf{T}}\mathbf{z}_i\right)\right)\right], \tag{6.4.5}$$

where $\mathbf{b} = [\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(m-1)}]^\mathsf{T}$, $\mathbf{b}^{(k)} = [\alpha^{(k)}, b_1^{(k)}, \ldots, b_p^{(k)}]^\mathsf{T}$, and
$\mathbf{z}_i = [1, \int \langle O_i(q_i, \gamma_i), \theta_1\rangle\, dt, \ldots, \int \langle O_i(q_i, \gamma_i), \theta_p\rangle\, dt]^\mathsf{T}$. Again, there is no analytical solution to this

optimization problem and since it is concave, we use the L-BFGS algorithm to solve. To use this method we need the partial derivative of the log-likelihood for each $\mathbf{b}^{(k)}$ which are,

$$\frac{\partial L_m(\mathbf{b})}{\partial \mathbf{b}^{(k)}} = \sum_{i=1}^{n} \left[y_i^{(k)} \mathbf{z}_i - \frac{1}{1 + \sum_{j=1}^{m-1} \exp\left(\mathbf{b}^{(j)\mathsf{T}} \mathbf{z}_i\right)} \exp\left(\mathbf{b}^{(k)\mathsf{T}} \mathbf{z}_i\right) \mathbf{z}_i \right].$$

To perform the overall minimization we alternate between finding the optimal warping functions and rotations $\{\gamma_i\}$ and $\{O_i\}$, and the optimal $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$. The algorithm for computing the optimal $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, $\{O_i^*\}$, and $\{\gamma_i\}$ is given in Algorithm 6.4.

Algorithm 6.4 Elastic Curve Multinomial Logistic Regression

1: Initialization Step: set $\{\gamma_i\} = \gamma_{id}$, $\{O_i^*\} = I$, and calculate SRVFs $\{q_i\}$, set $l = 1$.
2: **while** $\|L_{om}^{(l+1)} - L_{om}^{(l)}\|^2 < \epsilon$ **do**, where L_{om} is the log-likelihood
3: Find $\{\alpha^{*(j)}\}$ and $\{\beta^{*(j)}(t)\}$ using chosen basis and L-BFGS
4: Find $\{\gamma_i\}^{(l)}$ and $\{O_i\}^{(l)}$ using Algorithm 6.3 for each $i = 1, \ldots, n$
5: $l = l + 1$
6: **end while**
7: Compute the aligned SRVS using $\tilde{q}_i \mapsto O_i^*(q_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$ and aligned curves using $\tilde{\xi}_i = O_i^*(\xi_i \circ \gamma_i^*)$.

This procedure results in five items: $\{\alpha^{*(j)}\}$, $\{\beta^{*(j)}(t)\}$, $\{O_i^*\}$, $\{\gamma_i^*\}$, and $\{\tilde{\xi}_i\}$.

6.4.2 Prediction

After the model has been identified, one can use it to predict classes of future observation curves. Again, the answer is simple if the test data has not been observed with rotation and parameterization variability as the class probabilities would be computed using

$$P(y_i = k|q_i) = \phi(\alpha^{(k)} + \int_0^1 \left\langle q_i, \beta^{(k)} \right\rangle dt), \quad k = 1, \ldots, m, \tag{6.4.6}$$

and the class with maximum probability determines the class (i.e., $k^* = \arg\max_k P(y_i = k|q_i)$). However, a problem arises when the input has been observed with rotation and parameterization variability. The probability would be computed using

$$P(y_i = k|q_i, \gamma_i, O_i) = \phi(\alpha^{(k)} + \int_0^1 \left\langle O_i(q_i, \gamma_i), \beta^{(k)} \right\rangle dt), \quad k = 1, \ldots, m, \tag{6.4.7}$$

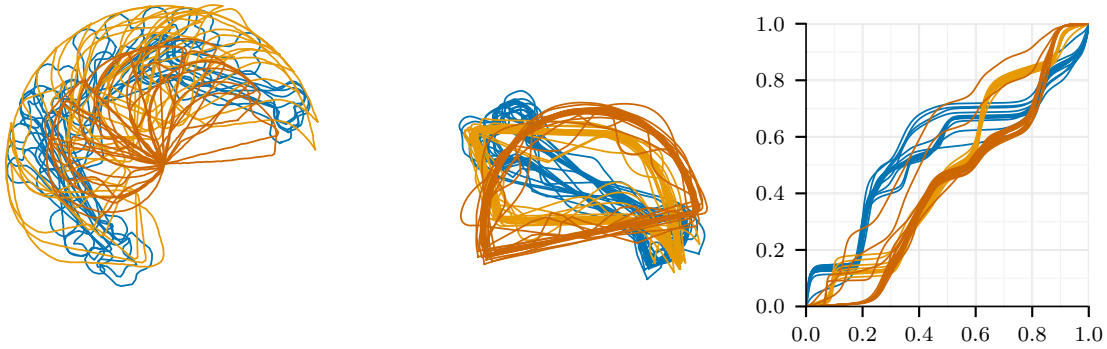
but the question remains on what O_i and γ_i should be.

The process is similar to that of elastic curve logistic regression. We propose the following: 1) Take the observed SRVF q_i and find the SRVF from the training set which has the smallest $\mathbb{L}^2$

distance $\|q_i - q_{train}\|^2$. 2) Determine the O_i and γ_i to be used for prediction as the corresponding time warping for q_{train} in the training sample. 3) Find the probability for each class, using Eqn. 6.4.7 with the corresponding chosen O_i and γ_i. 4) Determine the class membership using $k^* = \arg\max_k P(y_i = k|q_i, \gamma_i, O_i)$.

6.4.3 Experimental Results

In this section, we classify the curve data from the MPEG-7 database presented in Section 6.3.3. First we tested the model on distinguishing between the first three shapes (bone, half circle, and comma). Fig. 6.5a presents the original curves which demonstrate the differences in scale and rotation between the three shapes. For the estimation of the model we used a B-spline basis with 60 elements and estimated the parameters of the model using Algorithm 6.4. Fig. 6.5b presents the aligned curves in which we notice the curves have been aligned into three distinct groups. Fig. 6.5c presents the corresponding warping functions which also exhibit that the method registers the samples into three classes.



(a) $\{\xi_i\}$ (b) $\{\tilde{\xi}_i\}$ (c) $\{\gamma_i\}$

Figure 6.5: First three shapes from the MPEG-7 database, (a) un-registered curves, (b) registered curves, and (c) warping functions resulting from Elastic Curve Multinomial Logistic Regression.

To evaluate the performance of the algorithm, we performed a 5-fold cross-validation experiment to compare the classification of the elastic method versus the standard curve multinomial logistic regression (CMLoR). By standard we mean calculating the probability for each class using $P(y_i =$

$k|\xi_i) = \phi(\alpha^{(k)} + \int \langle \xi_i, \beta^{(k)} \rangle \, dt)$ and identifying the model parameters $\{\alpha^{(k)}\}$ and $\{\beta^{(k)}\}$ by solving

$$\underset{\{\alpha^{(k)}\},\{\beta^{(k)}\}}{\arg\max} \sum_{i=1}^{n} \left[\sum_{j=1}^{m-1} y_i^{(j)} [\alpha^{(j)} + \int \left\langle \xi_i, \beta^{(j)} \right\rangle \, dt] - \log(1 + \sum_{j=1}^{m-1} \exp(\alpha^{(j)} + \int \left\langle \xi_i, \beta^{(j)} \right\rangle \, dt)) \right]$$

using L-BFGS and a basis representation. Prediction was performed for the elastic method as described above and since we have the labels for the curves, we can calculate the classification performance. Table 6.2 presents the mean probability of classification across the folds with the corresponding standard deviation in parentheses for the first three shapes in Panel a, the first five shapes (bone, half circle, comma, heart, and misk) in Panel b, and all 65 shape classes in Panel c.

Table 6.2: Mean probability of classification using 5-fold cross-validation and standard deviation in parentheses for MPEG-7 data using curve logistic regression.

	Mean (STD)
CMLoR Warped Data	0.725 (0.102)
Pre-Align CMLoR	0.473 (0.194)
Cluster CMLoR	0.580 (0.045)
Elastic CMLoR	**0.933 (0.082)**
Elastic 1-NN	1.000 (0.000)
Kendall 1-NN	0.300 (0.0612)

(a) First three shapes

	Mean (STD)
CMLoR Warped Data	0.603 (0.053)
Pre-Align CMLoR	0.434 (0.066)
Cluster CMLoR	0.443 (0.119)
Elastic CMLoR	**0.880 (0.051)**
Elastic 1-NN	1.000 (0.000)
Kendall 1-NN	0.190 (0.041)

(b) First five shapes

	Mean (STD)
CMLoR Warped Data	0.042 (0.014)
Pre-Align CMLoR	0.431 (0.146)
Cluster CMLoR	0.296 (0.121)
Elastic CMLoR	**0.626 (0.029)**
Elastic 1-NN	0.850 (0.015)
Kendall 1-NN	0.028 (0.008)

(c) Entire data set

We also compare against two alignment-based CMLoR methods, which are the same alignment-based methods as described in Section 6.3.3, except we replace CMLoR for CLoR. For all three data sets the elastic method outperforms the standard CMLoR and the two alignment-based CMLoR. In particular, the classification accuracy is around 90% for first two cases and 60% for the entire

data set. Moreover, this improvement has a lower standard deviation across the folds indicating that the model generalized well for the data. We also compared the methods against the same two metric-based 1-NN classifiers described in the Section 6.3.3. Again, the elastic distance outperforms the Elastic CLMoR in all cases, this is to be expected as our model is linear. However, we greatly outperform Kendall's distance in all studied cases. Overall, the classification rates are improved when the model accounts for the rotation and parameterization variability in the model.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Statistical analysis of function data is important in a wide variety of applications, arising in nearly every branch of science from speech processing, and geology to biology and chemistry. In this work, we have presented a comprehensive approach that solves the problem of registering and modeling functions in a joint, metric-based framework.

7.1 Discussion

We proposed three approaches to the phase-variability problem for component analysis and regression. Our first main idea is to use an elastic distance to separate the given functional data into phase and amplitude components, and to develop individual models for these components. The specific models use fPCA and imposition of either multivariate Gaussian or nonparametric models on the coefficients. We illustrated the strengths of these models in two ways: random sampling and model-based classification of functional data. In the case of classification, we consider applications involving handwritten signatures, motion data collected using iPhones, and SONAR signals. We illustrate the improvements in classification performance when the proposed models involving separate phase and amplitude components are used.

Using the elastic metric, we then proposed a joint alignment and component analysis of functional data. Specifically, we incorporated the phase-variability into the objective function of the component analysis which in turn alters how the the optimization is performed. We then are able to extract the corresponding components and warping functions that align the functions. The solution is a more natural approach to the analysis as the alignment is not a "pre-processing" step, but rather part of the complete solution using one metric. We developed a joint alignment and fPCA as well as a joint alignment and fPLS and considered applications for activity recognition data.

Finally, we have proposed a new functional linear regression approach that addresses the problem of registering and modeling functions in one elastic-framework. We then extended the linear model to the case of functional logistic and functional multinomial logistic regression for the clas-

sification of functions. The strengths of these model are illustrated in two ways: the alignment of the functions and the classification of functional data. In the case of classification, we consider applications in physiological signals. We illustrate the improvements in classification performance when the proposed models involving phase-variability are used. We also provide an extension to $\mathbb{R}^2$ from $\mathbb{R}$ for the case of using curves as predictors and extend the linear, logistic, and multinomial logistic model to $\mathbb{R}^2$ and show results using shape data.

7.2 Future Work

As the main contribution of this work is functional data analysis using an elastic framework, we focus on that direction for future work. We provide a short list of open problems that still remain to be addressed:

1. The next natural step from elastic fPLS is to elastic functional canonical correlation analysis (fCCA). The difference between the two methods is the denominator in the cost function. In the fPLS method the denominator is just the norm of the weight functions, which guarantees the weight functions are unit norm. In fCCA the goal is to maximize the correlation instead of the covariance, so the denominator is the square-root of the variance of $r_{f,i} = \left\langle (q_{f,i}, \gamma_i), w_{q_f} \right\rangle$ multiplied by the square root of the variance of $r_{g,i} = \left\langle (q_{g,i}, \gamma_i), w_{q_g} \right\rangle$. This adds complication to the objective function, as now the optimization over $\{\gamma_i\}$ is in the denominator as well. As CCA is a popular method in multivariate statistics, and functional methods have been developed [42]; an extension to the inclusion of warping is the next logical step.

2. We have defined the estimation methods for the functional linear regression, as well as the logistic and multinomial logistic cases. The next step, is to study the consistency of the estimation algorithms and the rate of convergence. Moreover, an analysis of the estimated coefficient function, $\beta(t)$ and how to interpret it is needed.

3. The estimation methods for the elastic regression models are based on gradient-based approaches. Recently, Cheng [12] developed a Bayesian alignment method using the SRSF and a similar approach should be developed for the identification of the regression model parameters.

4. We have only extended the regression models for open curves in $\mathbb{R}^2$. These models need to be extended to handle open and closed curves in $\mathbb{R}^n$.

5. The developed methods have been only for 1-dimensional functional data, with an extension of the regression models for 2-dimensional curves. In the area of statistical analysis of curves,

there is desire for generative models and performing various component analysis. Extension of elastic fPCA and fPLS to open and closed curves in $\mathbb{R}^n$ opens up many possibilities for research and applications.

REFERENCES

[1] S. Amari. *Differential Geometric Methods in Statistics*. Lecture Notes in Statistics, Vol. 28. Springer, 1985.

[2] M. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vision*, 61:139–157, 2005.

[3] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[4] A. Bhattacharya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[5] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior. The relation between the ROC curve and the CMC. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 15 – 20, Oct. 2005.

[6] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010.

[7] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

[8] T. Cai and P. Hall. Prediction in functional linear regression. *Annals of Statistics*, 34(5): 2159–2179, 2006.

[9] H. Cardot, F. Ferraty, and P. Sarda. Functional linear model. *Statistics & Probability Letters*, 45:11–22, 1999.

[10] H. Cardot, F. Ferraty, and P. Sarda. Spline estimator for the functional linear model. *Statistica Sinica*, 13:571–591, 2003.

[11] N. N. Čencov. *Statistical Decision Rules and Optimal Inferences*, volume 53 of *Translations of Mathematical Monographs*. AMS, Providence, USA, 1982.

[12] W. Cheng, I. L. Dryden, and X. Huang. Bayesian registration of functions and curves. *arXiv:1311.2105 [stat.ME]*, 2013. URL http://arxiv.org/abs/1311.2105.

[13] G. E. Christensen and H. J. Johnson. Consistent image registration. *IEEE Trans. Medical Imaging*, 20(7):568–582, 2001.

[14] A. Cuevas, M. Febrero, and R. Fraiman. Linear functional regression; the case of fixed design and functional response. *The Canadian Journal of Statistics*, 30:285–300, 2002.

[15] B. Efron. Defining the curvature of a statistical problem (with applications to second order efficiency). *Annals of Statistics*, 3:1189–1242, 1975.

[16] T. Gasser and H. G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scandinavian Journal of Statistics*, 11:171–185, 1984.

[17] J. Gertheiss, A. Maity, and A.-M. Staicu. Variable selection in generalized functional linear models. *Stat*, 2(1):86–101, 2013.

[18] D. Gervini. Warped functional regression. *arXiv:1203.1975 [stat.ME]*, 2013. URL http://arxiv.org/abs/1203.1975.

[19] D. Gervini and T. Gasser. Self-modeling warping functions. *Journal of the Royal Statistical Society, Ser. B*, 66:959–971, 2004.

[20] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23): e215–e220, June 2000. URL http://www.physionet.org.

[21] P. Hall and J. L. Horowitz. Methodology and convergence rates for functional linear regression. *Annals of Statistics*, 35(1):70–91, 2007.

[22] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[23] G. M. James. Generalized linear models with functional predictors. *Journal of the Royal Statistical Society: Series B*, 64:411–432, 2002.

[24] G. M. James. Curve alignments by moments. *Annals of Applied Statistics*, 1(2):480–501, 2007.

[25] G. M. James, J. Wang, and J. Zhu. Functional linear regression that's interpretable. *Anals of Statistics*, 37(5A):2083–2108, 2009.

[26] S. Jeannin and M. Bober. Shape data for the mpeg-7 core experiment ce-shape-1, 1999. URL http://www.dabi.temple.edu/~shape/MPEG7/dataset.html.

[27] S. H. Joshi, E. Klassen, A. Srivastava, and I. H. Jermyn. A novel representation for riemannian analysis of elastic curves in $\mathbb{R}^n$. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, 2007.

[28] S. Jung, I. L. Dryden, and J. S. Marron. Analysis of principal nested spheres. *Biometrika*, 99 (3):551–568, 2012.

[29] S. G. Kargl, K. L. Williams, T. M. Marston, J. L. Kennedy, and J. L. Lopes. Acoustic response of unexploded ordnance (UXO) and cylindrical targets) and cylindrical targets. *Proc. of MTS/IEEE Oceans 2010 Conference*, pages 1 –5, 2010.

[30] R. E. Kass and P. W. Vos. *Geometric Foundations of Asymptotic Inference*. John Wiley & Sons, Inc., 1997.

[31] D. Kaziska. Functional analysis of variance, discriminant analysis and clustering in a manifold of elastic curves. *Communications in Statistics*, 40:2487–2499, 2011.

[32] D. Kendall. Shape manifolds, Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, Mar. 1984.

[33] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification/clustering homepage, 2001. URL www.cs.ucr.edu/~eamonn/time_series_data/.

[34] A. S. Khwaja, L. Ferro-Famil, and E. Pottier. SAR raw data simulation using high precision focusing methods. *European Radar Conference*, pages 33–36, 2005.

[35] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2004.

[36] A. Kneip and T. Gasser. Statistical tools to analyze data representing a sample of curves. *Annals of Statistics*, 20:1266–1305, 1992.

[37] A. Kneip and J. O. Ramsay. Combining registration and fitting for functional models. *Journal of the American Statistical Association*, 103(483), 2008.

[38] I. Koch, P. Hoffmann, and J. S. Marron. Proteomics profiles from mass spectrometry. *Special Section, Electronic Journal of Statistics*, in review, 2014.

[39] S. Kurtek, E. Klassen, Z. Ding, S. W. Jacobson, J. L. Jacobson, M. J. Avison, and A. Srivastava. Parameterization-invariant shape comparisons of anatomical surfaces. *IEEE Trans. Medical Imaging*, 30(3):849–858, 2011.

[40] S. Kurtek, A. Srivastava, and W. Wu. Signal estimation under random time-warpings and nonlinear signal alignment. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2011.

[41] S. Kurtek, A. Srivastava, E. Klassen, and Z. Ding. Statistical modeling of curves using shapes and related features. *Journal of the American Statistical Association*, 107(499):1152–1165, 2012.

[42] S. E. Leurgans, R. A. Moyeed, and B. W. Silverman. Canonical correlation analysis when the data are cuves. *Journal of the Royal Statistical Society, Series B*, 55(3):725–740, 1993.

[43] X. Liu and H. G. Müller. Functional convex averaging and synchronization for time-warped random curves. *Journal of the American Statistical Association*, 99:687–699, 2004.

[44] C. McCall, K. Reddy, and M. Shah. Macro-class selection for hierarchical K-NN classification of inertial sensor data. *Proc. of PECCS 2012*, Feb 2012.

[45] P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.*, 8:1–48, 2006.

[46] A. Mordecai. *Nonlinear Programming: Analysis and Methods*. Dover Publishing, 2003.

[47] H. G. Müller and U. Stadtmüller. Generalized functional linear models. *Annals of Statistics*, 33(2):774–805, 2005.

[48] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 10: 186–190, 1964.

[49] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[50] J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society, Ser. B*, 53(3):539–572, 1991.

[51] J. O. Ramsay and X. Li. Curve registration. *Journal of the Royal Statistical Society, Ser. B*, 60(2):351–363, 1998.

[52] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2005.

[53] C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of Calcutta Mathematical Society*, 37:81–91, 1945.

[54] P. Reiss and R. Ogden. Functional principal component regression and functional partial least squares. *Journal of American Statistical Association*, 102(479):984–996, 2007.

[55] D. Robinson. *Functional analysis and partial matching in the square root velocity framework*. PhD thesis, Florida State University, 2012.

[56] L. M. Sangalli, P. Secchi, S. Vantini, and V. Vitelli. Functional clustering and alignment methods with applications. *Communications in Applied and Industrial Mathematics*, 1(1): 205–224, 2010.

[57] L. M. Sangalli, P. Secchi, S. Vantini, and V. Vitelli. k-mean alignment for curve clustering. *Computational Statistics and Data Analysis*, 54:1219–1233, 2010.

[58] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. London:Chapman & Hall/CRC, 1998.

[59] M. Soumekh. *Synthetic Aperture Radar Signal Processing*. Wiley, 1999.

[60] A. Srivastava and I. H. Jermyn. Looking for shapes in two-dimensional, cluttered point clouds. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(9):1616–1629, 2009.

[61] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning and testing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):590–602, 2005.

[62] A. Srivastava, E. Klassen, S.H. Joshi, and I.H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011.

[63] A. Srivastava, W. Wu, S. Kurtek, E. Klassen, and J. S. Marron. Registration of functional data using fisher-rao metric. *arXiv:1103.3817v2 [math.ST]*, 2011. URL http://arxiv.org/abs/1103.3817v2.

[64] H. Tagare, D. Groisser, and O. Skrinjar. Symmetric non-rigid registration: A geometric theory and some numerical techniques. *Journal of Mathematical Imaging and Vision*, 34(1):61–88, 2009.

[65] R. Tang and H. G. Müller. Pairwise curve synchronization for functional data. *Biometrika*, 95 (4):875–889, 2008.

[66] W. S. Torgerson. *Theory and methods of scaling*. Wiley: New York, 1958.

[67] J. D. Tucker and A. Srivastava. Statistical analysis and classification of acoustic color functions. *Proc SPIE*, 8017:O1–Q10, April 2011.

[68] J. D. Tucker, W. Wu, and A. Srivastava. Generative models for functional data using phase and amplitude separation. *Computational Statistics and Data Analysis*, 61:50–66, 2013.

[69] J. D. Tucker, W. Wu, and A. Srivastava. Analysis of signals under compositional noise with applications to sonar data. *IEEE Journal of Oceanic Engineering*, 39(2):318–330, 2014.

[70] J. D. Tucker, W. Wu, and A. Srivastava. Analysis of proteomics data: phase amplitude separation using extended fisher-rao metric. *Special Section, Electronic Journal of Statistics*, in press, 2014.

[71] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa. Rate-invariant recognition of humans and their activities. *IEEE Transactions on Image Processing*, 8(6): 1326–1339, 2009.

[72] G. S. Watson. Smooth regression analysis. *Sankhya, Series A*, 20:101–116, 1964.

[73] H. Wold. Estimation of principal components and related models by iterative least squares. In P. Krishnaiaah, editor, *Multivariate Anlaysis*, pages 391–420. Academic Press: New York, 1966.

[74] W. Wu and A. Srivastava. An information-geometric framework for statistical inferences in the neural spike train space. *Journal of Computational Neuroscience*, 31:725–748, 2011.

[75] D. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. First international signature verification competition. *Proceedings of the Int. Conf. on Biometric Authentication*, pages 16–22, 2004.

[76] L. Younes. Computable elastic distance between shapes. *SIAM Journal of Applie Mathematics*, 58(2):565–586, 1998.

[77] L. Younes, P. W. Michor, J. Shah, D. Mumford, and R. Lincei. A metric on shape space with explicit geodesics. *Matematica E Applicazioni*, 19(1):25–27, 2008.

[78] F. Zhou and F. de la Torre. Canonical time warping for alignment of human behavior. *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2286–2294, 2009.

BIOGRAPHICAL SKETCH

J. Derek Tucker received his B.S. in Electrical Engineering Cum Laude and M.S. in Electrical Engineering from Colorado State University in 2007 and 2009, respectively. Upon completion of these degrees, he began working as a Research Scientist at the Naval Surface Warfare Center Panama City Division in Panama City, FL. In the Fall of 2011, he began his Ph.D. studies in the Statistics department at Florida State University under the co-advisement of Dr. Anuj Srivastava and Dr. Wei Wu. During this time his research has focused on functional data analysis, shape analysis, regression-type problems, and classification. He defended his PhD dissertation in May, 2014 and will be beginning a new career as a Senior Research Scientist at Sandia National Laboratories in Albuquerque, NM, in June.