Title
Bayesian Curve Registration and Warped Functional Regression

Author
Wang, Lu

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Bayesian Curve Registration and Warped Functional Regression

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Biostatistics

by

Lu Wang

2018

ABSTRACT OF THE DISSERTATION

Bayesian Curve Registration and
Warped Functional Regression

by

Lu Wang
Doctor of Philosophy in Biostatistics
University of California, Los Angeles, 2018
Professor Donatello Telesca, Chair

Functional data usually consist of a sample of functions, with each function observed on a discrete grid. The key idea of functional data analysis is to consider each function as a single, structured object rather than a collection of data points. To represent and investigate functional data, curve registration and functional regression are two important techniques. Curve registration is used to align random curves that display time variations. This procedure, known as functional convex averaging, leads to phase-variance adjusted mean functions. Therefore, compared to a simple averaged mean function, phase-variance adjusted mean function by functional convex averaging is a more accurate representation of the inherent function from which the functional data arise. Several curve registration methods are reviewed in this work, including landmark, self-warping and Bayesian hierarchical curve registration (BHCR). For BHCR, when the number of random curves is large or the sampling grid is intensive, the computational cost increases dramatically. To solve this problem, we introduce an accelerated BHCR algorithm via a predictive process model (PPM), known as PPM-BHCR. Tested by a simulation study and real data, this new method is demonstrated to save large amounts of computing time, without a large sacrifice of accuracy.

Functional regression is used to explore the relationship between the outcome and the predictor, where either or both of them are functional. In this work, several functional regression methods are reviewed according to the function-on-scalar, scalar-on-function and function-on-function categories. Registration is traditionally performed as a data preprocessing step before regression. In

this work, we introduce a new method called warped functional regression (WFR), which integrates curve registration and functional regression into one joint model. Therefore, we are able to provide prediction based on an unwarped predictor using this new model. The proposed method is evaluated by simulation studies and demonstrates high accuracy. Several case studies illustrate the key contributions of the proposed method in addressing complex scientific questions.

The dissertation of Lu Wang is approved.

Xiao Hu

Gang Li

Catherine Ann Sugar

Donatello Telesca, Committee Chair

University of California, Los Angeles

2018

To my dearest family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2007	B.S. Bioinformatics, Zhejiang University Hangzhou, Zhejiang, China.
2009	M.S. Bioinformatics, North Carolina State University Raleigh, North Carolina.
2009 - present	Ph.D. candidate in Biostatistics, University of California, Los Angeles, California.

CHAPTER 1

Introduction

Scientific experiments and economical activities have brought rapidly growing amounts of functional data. Functional data usually consist of a sample of functions, and each function is sampled on a discrete grid. The key idea of functional data analysis is to consider each function as a single, structured object rather than a collection of data points. This allows for building complex models to simultaneously explore the relationship within and between functions. In the past decades, researchers have proposed a series of functional data analysis methods with significant applications in clinical research, imaging technology, econometrics and many other emerging areas. For example, [Goldsmith et al., 2012] used penalized functional regression to relate intracranial, white matter tracts to cognitive disability in multiple-sclerosis patients. Furthermore, [Sood et al., 2009] predicted the market penetration of new products using augmented functional regression.

Several perspectives on functional data analysis are to be considered when we survey its methodology and application. The first perspective, which is also a prerequisite for functional regression, is curve registration. The visualized representation of functional data often exhibits time variations, in the sense that they have similar shapes, but different phases. Therefore, it is necessary to synchronize the curves for the purpose of graphic representation or further formal inference. This process is known as curve registration, which involves transforming the time argument t so that curves are aligned. Existing curve registration methods include landmark registration, the self-warping function and Bayesian hierarchical curve registration (BHCR). The MCMC sampling process used in BHCR provides satisfactory estimation accuracy; however, it is time consuming. It is especially problematic when the number of curves or sampling points in each curve is large. In Chapter 2, we introduce the predictive process model-based BHCR (PPM-BHCR) to solve this problem. The simulation study demonstrates that the new method significantly accelerates BHCR

and does not sacrifice accuracy much. We also provide an example of using this new method to register intracranial pressure curves from neurosurgical patients.

Functional regression is a powerful tool for investigating the relationships among functional and scaler data. Depending on the types of outcomes and predictors, functional regression can be categorized into three classes: functional predictor regression (scalar-on-function), functional response regression (function-on-scalar) and function-on-function regression. These methods are reviewed in detail in the rest of this chapter. Although much literature and work has been focused on the area of functional regression, most of it treats functional regression as a separate step from curve registration. In Chapter 3, we propose warped functional regression (WFR) under the Bayesian framework to incorporate registration as an intrinsic part of the regression model, and we use the function-on-function type of functional regression to demonstrate this method. The proposed method is evaluated by a simulation study, and it is applied to two case studies for demonstration purposes.

The rest of this chapter reviews the fundamentals of functional data smoothing and the different methods for curve registration. We also review the three types of functional regression, and we then demonstrate Bayesian functional regression using two case studies.

1.1 Smoothing of Functional Data

1.1.1 Property of functional data

Functional data are usually recorded in the form $\boldsymbol{y} = (y(t_1), ..., y(t_j), ...y(t_m))$, where $y(t_j)$ is the observation of an unknown function f at time t_j, and it can be simply written as $y(t)$ for general purpose. Time is the most commonly used continuum, but it can also be other kind of scale, like spatial position. Instead of considering $\boldsymbol{y} = (y(t_1), ..., y(t_m))$ as a series of individual observations, the basic assumption of functional data analysis is to consider $\boldsymbol{y}$s as a complete set of reflection of the unknown function f. Hence the functional observation $\boldsymbol{y}$ can be denoted as $f(t)$ plus some measurement error. This unknown function f is what we want to investigate. To estimate f we need to make another fundamental assumption, that is the underlying function f is smooth in the

space of t so that it can be approximated by certain combination of known continuous functions. The data set in functional data analysis is often consisted of a group of random curves, where the curve itself has certain practical meaning, and it is more reasonable to treat each curve as an entity instead of treating each observation at every time point as an entity. For example, the growth data where height is a function of age, and weather data where temperature is a function of time. Figure 1.1 shows the heights of 54 girls measured from age 1 to 18 in Berkeley Growth Study ([Tuddenham and Snyder, 1954]). By visually examining the growth curves, it is clear that during age 8 to 12 the height growing speed is relatively higher than during age 14 to 18. The replication of the growth curves help us find out such height growing trend.

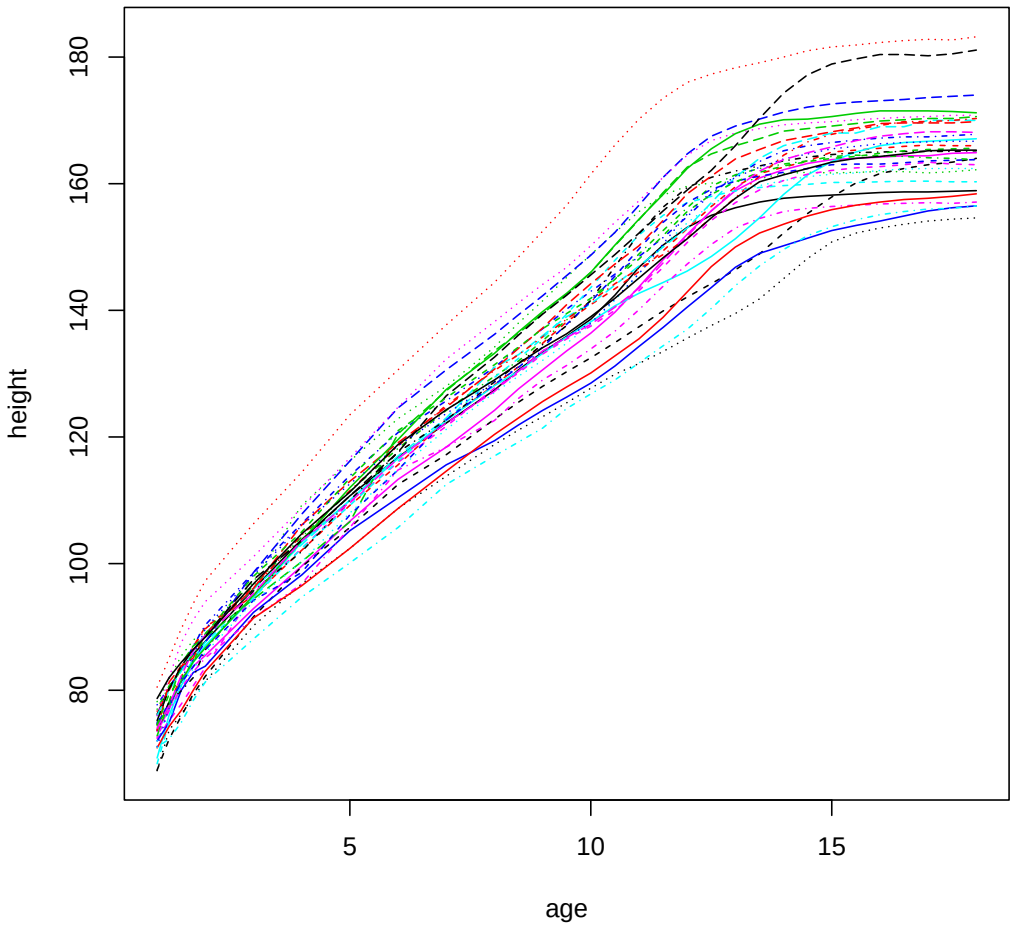

Figure 1.1: Berkeley Growth Data. Heights of 54 girls measured from age 1 to 18.

Functional data analysis can be categorized into exploratory analysis and confirmatory analysis. In exploratory functional data analysis, people concentrate on visualizing functional data and make descriptive statement about trend in data. Two major techniques in this category are curve registration and functional principle component analysis. In the confirmatory functional data analysis, people are more interested in using functional data to explain or predict relevant outcome. This task can be accomplished by functional regression model. Before entering detailed description of these methods, we will first discuss basic technique of smoothing functional data, which is the preliminary step of any type of functional data analysis.

1.1.2 Basis systems

Since the unknown function f is assumed to be smooth, it can be denoted by a linear combination of known functions:

$$f(t) = \sum_{k=1}^{K} \beta_k b_k(t). \tag{1.1}$$

Here $b_k(t)$ is called basis function. In a set of basis functions, $b_k(t)'s$ are independent with each other. There are many options for the choice of basis function. Ideally basis function should have properties that match the functional data so that only a small number K of basis functions is needed to well approximate $f(t)$. For periodic functional data, Fourier basis system is popular. It is a set of trigonometric functions $(\sin\omega t, \cos\omega t, \sin 2\omega t, \cos 2\omega t, ...)$. Wavelet basis system is another popular choice for periodic data. Compared to Fourier basis, wavelet has better performance when dealing with rapid change in random curves. By adopting the Discrete Wavelet Transformation, estimation of basis function coefficients is much faster for wavelet than Fourier basis system.

For non-periodic functional data, spline basis functions are often used. For example, the truncated power series $(1, t^1, t^2, t^3, (t-\xi_1)^3_+, ..., (t-\xi_k)^3_+)$. In this kind of basis system, $\xi_1, ..., \xi_k$ are called *knots*. They are points within the range of t and divide the range into subintervals. In each subinterval different splines are fit to approximate $f(t)$. Because of this flexibility, only a relative small number of spline functions is needed. The number and location of knots determine how accurate the approximation can be. Later on we will discuss how to determine these two factors. Each spline function is a polynomial of order m, which is one more than the highest power of the

polynomial. Adjacent splines join smoothly at knot. With k interior knots and cubic polynomial, $k + 4$ basis functions are needed for the approximation. The most popular spline basis function is B-spline basis function ([de Boor, 1978]) because of its good computational property. An order m B-spline basis function is positive over no more than m adjacent subintervals, so that the inner product matrix of basis functions is banded with nonzero values, and the rest large portion of the matrix is zero valued. This property leads to the result that even the number of basis functions increase to a very large value, the calculation of the inner product of design matrix is still manageable.

1.1.3 Smoothing functional data

As we mentioned: each functional observation arises from a curve or a function; the curve being estimated is smooth, which means it can be well approximated by a linear combination of basis functions:

$$\begin{aligned} y(t) &= f(t) + \varepsilon \\ &= \sum_{j=1}^{K} \beta_k b_k(t) + \varepsilon \end{aligned} \tag{1.2}$$

where ε is white noise. $\boldsymbol{f} = \{f(t_j)\}$ can also be expressed in matrix format: $\boldsymbol{f} = \mathbf{B}\boldsymbol{\beta}$, where $\mathbf{B}$ is the design matrix of basis functions evaluated at different time points. Smoothing functional data is done by estimating the coefficient vector $\boldsymbol{\beta}$. A simple way to do this is to use ordinary least square (OLS) fit:

$$\hat{\boldsymbol{\beta}} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\boldsymbol{y} \tag{1.3}$$

So the smoothed functional data are:

$$\hat{\boldsymbol{y}} = \mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\boldsymbol{y} \tag{1.4}$$

OLS estimation is lack of control over the degree of smoothing and local fluctuation of curve. The estimated function is often over-smoothed. A more sufficient method should provide appropriate degree of smoothing and not miss significant local fluctuation. These two conflicted properties are balanced by choosing number of knots and their locations carefully. Such methods are introduced below.

1.1.4 Smoothing functional data by roughness penalty

The major challenge of smoothing functional data is how to decide the number of knots and their locations. Authors proposed two categories of methods to solve this problem. One is called smoothing splines: it avoids selecting knots by using a maximal set of knots and imposing roughness penalization ([Hastie and Tibshirani, 1990]). The criterion to minimize is:

$$RSS(f,\lambda) = \sum_{j=1}^{m}[y(t_j) - f(t_i)]^2 + \lambda \int_T [f''(t)]^2 dt \tag{1.5}$$

where λ is the tuning parameter which controls smoothness. Larger λ brings smoother curve, and smaller λ brings more wiggly curve. It can be shown that 1.5 has an explicit unique minimizer and that minimizer is a natural cubic spline with knots at unique values of x_i:

$$\hat{\boldsymbol{\beta}} = (\mathbf{B}'\mathbf{B} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{B}'\boldsymbol{y}, \tag{1.6}$$

where $\Omega_{ij} = \int B_i''(t)B_j''(t)dx,\ B_{ij} = B_j(t_i)$. Computing $\boldsymbol{\Omega}$ is usually done by numerical integration. In practice it seldom requires very high accuracy. The tuning parameter λ can be found by cross-validation. Cross-validation is common in a wide range of statistic problems. Particularly for smoothing spline, the generalized cross-validation (GCV) developed by [Craven and Wahba, 1978] is popular. GCV avoids partitioning the data into training set and validation set. The tuning parameter λ is chosen by calculating the following criterion:

$$\text{GCV}(\lambda) = \frac{n\text{SSE}}{[\text{trace}(\mathbf{I} - (\mathbf{B}'\mathbf{B} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{B}')]^2} \tag{1.7}$$

GCV is not only more efficient than ordinary cross-validation but also more reliable in the sense of being less possible to be under-smoothing.

1.1.5 Smoothing functional data by knot selection

Smoothing splines method has the advantage of computational easiness and simplicity to control the smoothness. The drawback is also obvious: since it uses a global penalization parameter it is lack of flexibility to capture local features when dealing with curve with inhomogeneous smoothness. It can be over-smoothed in some area and under-smoothed elsewhere. Hence another class of

methods are proposed: regression splines, which does select knots and their locations. [Friedman, 1991] and [Luo and Wahba, 1997a] proposed stepwise selection of knots which adopted traditional stepwise/backward/forward variable selection methods. However these methods suffer the inherent drawbacks of stepwise selection (inappropriate use of test and p-value, biased regression coefficients and confounding problem). To improve, [Osborne et al., 1998] proposed knot selection via LASSO and an algorithm that allows efficient calculation of estimation. [Zhou and Shen, 2001] proposed the adaptive regression spline method: knot insertion is through guided search that less smooth area tends to have more knot placed; spline is fitted locally with neighborhood size defined by the spline used.

From Bayesian perspective, [Smith and Kohn, 1996] proposed a knot selection and estimation method based on the Bayesian variable selection method by [George and Mcculloch, 1993]. Suppose we use cubic power truncated splines: $\left\{1, t^1, t^2, t^3, (t-\xi_1)_+^3, ..., (t-\xi_k)_+^3\right\}$. To facilitate the selection of knots, authors introduce the indicator variable $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_k)\prime$:

$$P(\gamma_j = 1) = 1 - P(\gamma_j = 0) = \omega. \tag{1.8}$$

When $\gamma_j = 0$, we remove the corresponding knot ξ_j from $\boldsymbol{\xi}$; when $\gamma_j = 1$, we add the corresponding knot ξ_j to $\boldsymbol{\xi}$. The number of knots currently existing in the spline can be calculated by $k\boldsymbol{\gamma} = \boldsymbol{\gamma}\prime\mathbf{1}$. So the current design matrix of basis functions $\mathbf{B}$ and $\boldsymbol{\beta}$ also depend on $\boldsymbol{\gamma}$, we denote them as $\mathbf{B}_\gamma$ and $\boldsymbol{\beta}_\gamma$. The sampling distribution of Y is:

$$\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2 \sim N(B\boldsymbol{\gamma}\boldsymbol{\beta}\boldsymbol{\gamma}, \sigma^2 I_n). \tag{1.9}$$

The next thing is to assign priors on parameters $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and σ^2. Assuming σ^2 and $\boldsymbol{\gamma}$ are independent with each other, we use inverse Gamma as the prior of σ^2,

$$\sigma^2 \sim IG\left(\nu/2, \nu\lambda/2\right). \tag{1.10}$$

where the shape and scale parameters are prespecified small values. Assuming $\gamma_j' s$ are independent with each other, authors use the Bernoulli distribution as the prior for each of the elements of $\boldsymbol{\gamma}$:

$$\gamma_j|\omega \sim Bernoulli(\omega) \tag{1.11}$$

and set ω to be 0.5 which represents no prior knowledge about whether a variable is included or not (In next section we will show that actually Beta distribution can be used here as the prior for ω, and in the joint prior distribution of ω and $\boldsymbol{\gamma}$, ω can be integrated out). For $\boldsymbol{\beta}$, we assign the Zellner's g-prior on it:

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{\gamma} \sim N(0, c\sigma^2(\mathbf{B}_\gamma'\mathbf{B}_\gamma)^{-1}) \tag{1.12}$$

where c is a positive scale constant specified by user. Since the priors are all conjugate, their full conditional posteriors are in closed form:

$$\begin{aligned} \boldsymbol{\beta}|\sigma^2, \boldsymbol{\gamma}, \boldsymbol{y} &\sim N\left(((1+c)\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\boldsymbol{y}, \sigma^2(\mathbf{B}'\mathbf{B} + c\mathbf{B}'\mathbf{B})^{-1}\right) \\ \sigma^2|\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{y} &\sim IG\left(\frac{n+\nu}{2}, \frac{|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}|^2 + \nu\lambda}{2}\right) \\ P(\gamma_j = 1|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y}) &= \frac{a}{a+b} \end{aligned} \tag{1.13}$$

where $a = f(\boldsymbol{\beta}|\boldsymbol{\gamma}_{-j}, \gamma_j = 1)\omega,\ b = f(\boldsymbol{\beta}|\boldsymbol{\gamma}_{-j}, \gamma_i = 0)(1-\omega)$. Gibbs sampler is used to sample from the full conditional posteriors. At each iteration the smoothing function $\hat{f(x)}$ is estimated by the sampled parameters. After retrieving a sufficient long sequence of $\hat{f(x)}$, the final estimation of the smoothing function can be calculated by:

$$\hat{f}(t) = \frac{1}{S}\sum_{s=1}^{S} f(t)^{(s)} \tag{1.14}$$

This average of samples converges to the real mean of $f(t)$.

A more adaptive method of curve-fitting with knot selection is proposed by [Denison et al., 1998]. They use free-knot splines that both the number and location of knots were treated as random variables. [Denison et al., 1998] avoid specifying a prior on $\boldsymbol{\beta}$ by plugging in its least square estimate. [DiMatteo et al., 2001] improved this method by assigning Gaussian prior on $\boldsymbol{\beta}$ and approximate the likelihood ratio by Bayesian information criterion (BIC). This improved method has certain advantages, as will be discussed below, but we first describe the general scheme of the free-knot curve-fitting.

In addition to the polynomial coefficient $\boldsymbol{\beta}$ and variance σ^2, we introduce two more random parameters: k, the number of knots for current spline, and $\boldsymbol{\xi} = (\xi_1, ..., \xi_k)$, the location of these

knots, where $a < t_{(1)} < \xi_1 < ... < \xi_k < t_{(n)} < b$. For k we can adopt a Poisson prior or discrete uniform prior on $1, 2, ..., K$. In practice, the results are not very sensitive to the choice between these two priors. Given k, the prior of $\boldsymbol{\xi}$ is Dirichlet distribution by scaling $[a, b]$ to $[0, 1]$, which is the multivariate generalization of Beta distribution. For $\boldsymbol{\beta}$ and σ^2, we use the same prior as in 1.10 and 1.12. Involving $\boldsymbol{\beta}$ in the posterior distribution certainly brings computational complication. This can be avoided by integrating out $\boldsymbol{\beta}$ and σ^2:

$$p(\boldsymbol{y}|k, \boldsymbol{\xi}) = \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, k, \boldsymbol{\xi})\pi(\boldsymbol{\beta}, \sigma^2|k, \boldsymbol{\xi})d\boldsymbol{\beta}d\sigma \tag{1.15}$$

The marginalized likelihood 1.15 can be well approximated by BIC. Since with changing k the dimension of $\boldsymbol{\xi}$ is also changing. Standard MCMC sampling methods do not apply to this case when the dimension of parameter space is changing. Authors propose to use the reversible jump MCMC sampling by [Green, 1995] which is designed to have Markov chain samplers jump between parameter spaces of different dimensionality.

There are three possible transitions for each sampling step: addition, deletion and relocation of knots, with following probabilities correspondingly:

$$b_k = c\ min1, p(k+1)/p(k), \quad d_k = c\ min1, p(k-1)/p(k), \quad \eta_k = 1 - b_k - d_k. \tag{1.16}$$

These probabilities ensure detailed balance by $b_k p(k) = d_{k+1} p(k+1)$. After deciding the type of next transition, the proposal and sampling rule of each type of step are defined as below:

Birth step. Propose a new knot ξ^* based on current knots $\boldsymbol{\xi}_k$. First uniformly choose one knot from current knots, and then generate the new knot from a distribution $h(\xi^*|\boldsymbol{\xi}_k, \tau)$ centered at the chosen knot with certain spread parameter τ. The proposal probability is:

$$q(M_{k+1}|M_k) = b_k \frac{1}{k} \sum h(\xi^*|\boldsymbol{\xi}_k, \tau) \tag{1.17}$$

Death step. Uniformly choose one knot ξ^* from existing knots to delete. Then the proposal probability is:

$$q(M_{k-1}|M_k) = d_k \frac{1}{k} \tag{1.18}$$

Relocation step Uniformly choose one knot ξ^* from existing knots to relocate. The new location

of the knot is decided as in birth step. The proposal probability is:

$$q(M_{new}|M_{current}) = \eta_k \frac{1}{k} h(\xi^*|\boldsymbol{\xi}_k, \tau) \tag{1.19}$$

In [Denison et al., 1998]'s method, the acceptance probability is decided by

$$\alpha\text{=min(1,likelihood ratio} \times \text{prior ratio} \times \text{proposal ratio)}$$

which can be calculated for each of the three steps by formulas we have above. In [DiMatteo et al., 2001]'s method, they take one more step to also sample $\boldsymbol{\beta}$ given $(k, \boldsymbol{\xi})$ by importance reweighting. The sampled $\boldsymbol{\beta}$ is used to calculate the estimation function $\hat{f}(x)$. Comparing these two methods, [Denison et al., 1998] plugs least square estimate of $\boldsymbol{\beta}$ into the likelihood ratio, which makes the effect of knot number k vanish as dataset gets large. On the contrast, [DiMatteo et al., 2001] uses BIC which penalizes the likelihood ratio for dimensionality, which makes the approximation more accurate.

1.1.6 Smoothing functional data by Bayesian P-spline

Penalized regression splines introduced by Eilers and Marx (1996) is for univariate scatterplot smoothing. It assumes the unknown function $f(\cdot)$ can be approximated by a polynomial spline written in terms of a linear combination of B-spline basis functions. The smoothness of the spline is controlled by the number and location of knots. Small number of knots may lead to over-smoothed function, and large number of knots may result in under-smoothed function which is sensitive to local fluctuation. To balance between smoothness and flexibility, penalized regression splines use a moderate number of equally spaced knots and penalize on the magnitude of basis function coefficients. Such approach is called P-spline. Bayesian version of P-spline ([Lang and Brezger, 2004]) replaces the penalty by their stochastic analogues, i.e. Gaussian random walk priors. It is a common basic technique in Bayesian functional data analysis. Compared to the traditional P-spline, Bayesian P-spline is easier to be adaptive when $f(\cdot)$ is highly oscillating in local area, by using locally adaptive smoothing parameter instead of global smoothing parameter. Such extension has been introduced in [Lang et al., 2002], [Luo and Wahba, 1997b] and [Ruppert and

Carroll, 2000]. Bayesian P-spline is implemented through MCMC simulation. When the conjugate Gaussian prior is used, we use Gibbs sampler to sample from full conditionals of parameters.

For P-spline, suppose we choose a moderate number of knots and have p basis functions. The penalized likelihood is:

$$L = l(y, \beta_1, ..., \beta_p) - \lambda \sum_{l=k+1}^{p} (\Delta^k \beta_l)^2$$

where Δ^k is the kth order difference operator of adjacent basis function coefficients, λ is the parameter for control of smoothness. Therefore, the penalty is on finite differences of the coefficients of adjacent B-splines. It has the advantage of reducing number of differentiation operations from number of observations to number of basis functions minus k. Such penalty is closely related to the penalty in model 1.5 which penalizes on derivative of fitted function. Take the first order derivative as an example:

$$h \sum \beta_l B_l'(t; q) = - \sum \Delta\beta_{l+1} B_l(t; q-1) \tag{1.20}$$

where h is the space between knots, q is the degree of the B-spline, $\Delta\beta_l = \beta_l - \beta_{l-1}$.

In the Bayesian version of P-spline, $k = 1$ and $k = 2$ are corresponding to the first and second order random walk. The parameters are generated by the random process:

$$\beta_j = \beta_{j-1} + e_j \text{ or } \beta_j = 2\beta_{j-1} - \beta_{j-2} + e_j \tag{1.21}$$

with Gaussian error $e_j \sim N(0, \tau^2)$. For the first order random walk, $\beta_1 \propto const$. For the second order random walk, $\beta_1 \propto const$ and $\beta_2 \propto const$. Smoothness is controlled by the additional variance parameters τ^2's, which corresponds to λ in P-spline. So the precision matrices of the first order random walk and second order random walk priors on $\boldsymbol{\beta}$ are $\boldsymbol{\Omega}_1/\tau^2$ and $\boldsymbol{\Omega}_2/\tau^2$:

$$\boldsymbol{\Omega}_1 = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & \ddots & & \\ & -1 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 & \\ & & \ddots & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}$$

$$\boldsymbol{\Omega}_2 = \begin{pmatrix} 6 & -4 & 1 & & & & & \\ -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 5 & -2 \\ & & & & & 1 & -2 & 1 \end{pmatrix}$$

Blank cells are all 0s. Gamma prior is assigned on the smoothing parameter τ^2. For full Bayesian inference, the unknown variance parameter τ^2 is also considered as random and estimated simultaneously with β_j. Usually we assign inverse Gamma prior on τ^2. In some applications the choice of a global variance parameter τ^2 may not be adequate. For example, the underlying function is highly oscillating. In such situation, we replace τ^2 by τ^2/δ_j where Gamma prior is assigned on the hyperparameter δ_j. [Lang and Brezger, 2004] shows successful and comprehensive application of Baysian P-spline in simulation study and multiple real data analysis. Particularly, the amount of smoothing is estimated simultaneously with the unknown function, which is considered as a distinct advantage of Bayesian P-spline. In the following curve registration and functional regression chapter, we also choose to use Bayesian P-spline as our primary method for smoothing functional data.

1.2 Curve Registration

1.2.1 Introduction

Before proceeding to functional modeling, we discuss one critical problem: functional data often exhibit variation in both phase and amplitude, and confounding these two factors leads to severe problems. For example, the cross-sectional average does not reflect the real pattern when each individual peak comes in different timing. Hence synchronizing the curves is necessary before investigating the average curve. This process is called curve registration, an essential preliminary to functional data analysis. Fig. 1.2 (from [Kneip and Gasser, 1992]) shows a motivating example

of curve registration. Curve registration usually involves finding a time warping function that aligns all curves, and also estimating an average curve. In this chapter we will first discuss several important curve registration methods, and then present the Bayesian hierarchical curve registration developed by [Telesca and Inoue, 2008], which will be used for preprocessing functional data in chapter 4.

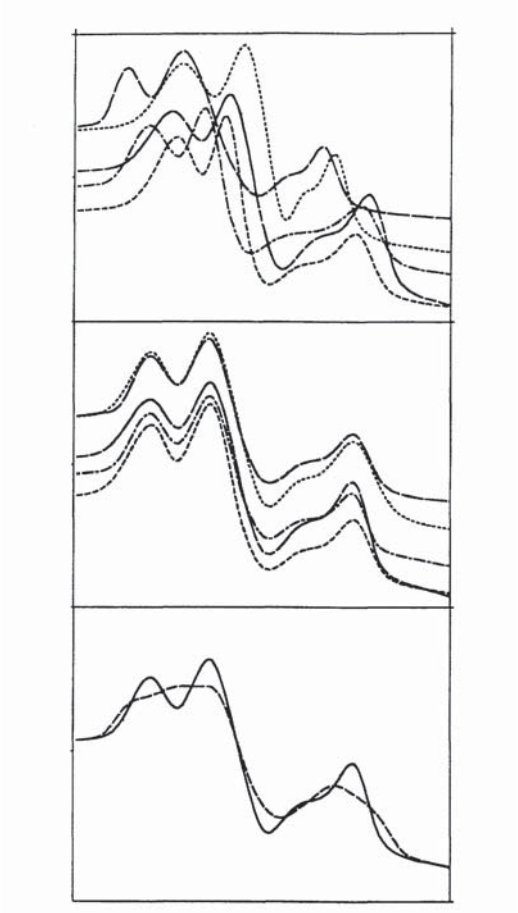

Figure 1.2: Top figure shows before curve registration curves have different phase and amplitude; middle figure shows after curve registration curves are aligned; bottom figure shows the average curve without registration (dashed line) and with registration (solid line).

1.2.2 Curve Registration Methods

1.2.2.1 Landmark Registration

[Ramsay and Silverman, 2005] provides a good summary of curve registration methods. The simplest way to align curves is to shift the curve by certain shifting parameter δ_i for the ith curve $x_i(t)$:

$$x_i^*(t) = x_i(t + \delta_i) \tag{1.22}$$

The shifting parameter is estimated by minimizing certain criterion. Let $\hat{\mu}(t)$ denote the estimated mean function. A global registration criterion is:

$$RSSE = \sum_{i=1}^{n} \int [x_i(t + \delta_i) - \hat{\mu(t)}]^2 dt$$

Our target function is the mean function $\hat{\mu(t)}$. By applying curve registration, it can be better estimated iteratively: begin with the unregistered mean function, calculate the shift parameter, update the mean function with registered curves. This procedure converges very fast, usually within three iterations.

More generally, instead of using the shifting parameter, we want to estimate a warping function $h_i(t)$ to align the curve:

$$x_i^*(t) = x_i[h_i(t)] \tag{1.23}$$

One possibility to estimate $h_i(t)$ is to first identify a specific feature or landmark for a curve, then align each curve according to that landmark. A landmark is usually an extrema like maxima, minima or zero crossing of the curve. [Kneip and Gasser, 1992] and [Gasser and Kneip, 1995] developed the landmark registration which refers to landmark as a structural feature. Searching for structure features is not easy due to individual dynamic variations and noise. [Gasser and Kneip, 1995] proposed to use frequencies of occurrence as the standard to identify structural features. They defined the distribution of extrema locations as structural intensity $f(t)$:

$$f(t) = \lim_{h \to 0} \frac{1}{2h} E\#M(r_i) \cap [t - h, t + h] \tag{1.24}$$

where r_i is the ith curve, $M(r_i)$ is the collection of extremas of r_i, $M(r_i) \cap [t - h, t + h]$ is the number of elements in $M(r_i)$ for the interval $[t - h, t + h]$, E is the expectation operator respect to

the sample of curves. The modes of $f(t)$ reflects the typical locations of structural features. $f(t)$ can be estimated through kernel density estimation, which involves kernel estimators for the curve function $r_i(t)$ and structural intensity. The bandwidth of kernel estimator for $r_i(t)$ is determined by "plug-in" bandwidth selector ([Gasser et al., 1991]). The bandwidth of kernel estimator for $f(t)$ is determined by a cross-validation type method suggested by [Rice and Silverman, 1991].

After the structural points are identified, a time-warping function $h(t)$ is determined to satisfy the following conditions:

1. Let $\tau_i = (\tau_{i1}, ..., \tau_{il})$ denote the l structural points for the ith curve, $\bar{\tau} = (\bar{\tau_1}, ..., \bar{\tau_l})$ denote the averaged structural points, $[a, b]$ is the support for all t.

2. $h(t)$ is a continuous function and strictly monotonically increasing for all $t \in [a, b]$.

3. $h_i(\bar{\tau}) = \tau_i$, align structural points to their average locations.

As an example, $h_i(t)$ can be determined by smooth, strictly monotonical interpolation of the points $(\tau_{i1}, \bar{\tau_1}), ..., (\tau_{il}, \bar{\tau_l})$.

1.2.2.2 Self-modeling Warping Function

The landmark registration is straightforward to interpret. However, it suffers from several undesirable aspects: some curves may have missing or ambiguous landmarks; variations of region outside structural area might be ignored; when sample size is large, identification of landmarks for each curve is time consuming. Instead of using landmarks, [Gervini and Gasser, 2004] proposed the self-modeling functions which is called continuous monotone registration without requiring landmarks. It assumes the sample curves $x_1(t), ..., x_n(t)$ follow the model:

$$x_i(t) = a_i \mu\{\nu_i(t)\} + \epsilon_i(t), \quad t \in T \subset \mathbb{R}, \;\; i = 1, ..., n \tag{1.25}$$

where $\mu : T \to \mathbb{R}$ is the structural mean, $\nu_i : T \to T$ is the monotone increasing function, and ϵ_i is random error. The warping function $w_i(t)$ defined as $w_i(t) = \nu_i(t)^{-1}$ is modeled as:

$$w_i(t) = t + \sum_{j=1}^{q} s_{ij} \phi_j(t) \tag{1.26}$$

The component $\phi_j(t)$ is modeled as:

$$\phi_j(t) = \boldsymbol{c}_j'\boldsymbol{\beta}(t) \tag{1.27}$$

where $\boldsymbol{\beta}(t) = (\beta_1(t), ..., \beta_p(t))$ is a vector of B-spline basis functions. Each $\phi_j(t)$ is considered as accounting for time variability in different segments of T. It is motivated by landmark registration that each $\phi_j(r)$ is associated with a "hidden landmark". To ensure the identifiability of ϕ-functions and other coefficients, certain conditions must be satisfied.

The parameters of 1.25 and 1.26 are estimated by minimizing the following criterion:

$$\begin{aligned} AISE &= \frac{1}{n}\sum_{i=1}^{n}\int [x_i(t) - a_i\mu\{\nu_i(t)\}]^2 dt \\ &= \frac{1}{n}\sum_{i=1}^{n}\int [x_i(w_i(t)) - a_i\mu(t)]^2 w_i'(t) dt \end{aligned} \tag{1.28}$$

The estimated $\mu(t)$ is:

$$\hat{\mu}(t) = \frac{\sum_{i=1}^{n}\hat{a}_i\hat{w}_i'(t)\hat{x}_i^*(t)}{\sum_{i=1}^{n}\hat{a}_i^2\hat{w}_i'(t)} \tag{1.29}$$

where $\hat{x}_i^*(t) = x_i\{\hat{w}_i(t)\}$. The coefficient vectors $\boldsymbol{a}$, $\boldsymbol{c}$ and $\boldsymbol{s}$ are estimated by Newton-Raphson method that they are updated sequentially. Notice here $x_i(t)$ is pre-smoothed. The number of components q and the number of basis functions p are determined by cross-validation. It is proved that under appropriate regularity conditions, the estimator $\hat{\boldsymbol{a}}$, $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{s}}$ are consistent as the number of curves goes to infinity.

After all, this self-modeling warping function avoids individual identification of landmarks and makes a more efficient use of data. It also provides flexibility of exploring the curve structure and in the mean time avoids over-fitting. Matlab programs on Dr. Gervini's web site are provided to implement this method. In next section, we will discuss the Bayesian curve registration ([Telesca and Inoue, 2008]) which adopts the self-modeling idea.

1.2.3 Bayesian Hierarchical Curve Registration

1.2.3.1 Model Formulation

Let $x_i(t)$ denote the ith observed curve for one subject at time t, $i = 1, .., N$, $t \in T = [a, b]$. BHCR is a three-stage hierarchical model.

Stage One.

$$\begin{aligned} x_i(t) &= m_i(\mu_i(t)) + \epsilon_i \\ m_i(t) &= c_i + a_i m(\mu_i(t); \boldsymbol{\beta}) = c_i + a_i \mathbf{B}'_m(\mu_i(t))\boldsymbol{\beta} \\ \mu_i(t) &= \mathbf{B}'_\mu(t)\boldsymbol{\phi}_i \end{aligned} \tag{1.30}$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$. $m(t; \boldsymbol{\beta})$ denotes a common shape function and $\mu_i(t)$ denotes a curve-specific time transformation function. $\mathbf{B}_m(t)$ and $\mathbf{B}_\mu(t)$ are vectors of basis functions. Here we use b-spline basis functions ([de Boor, 1978]) because of its computational advantages.

Stage Two. Given the common shape function $m(t; \boldsymbol{\beta})$, individual curves can vary by scale and level of response. We assign Gaussian priors to these two parameters as:

$$\begin{aligned} c_i &\sim N(c_0, \sigma_c^2) \\ a_i &\sim N(a_0, \sigma_a^2) I(a_i > 0) \end{aligned} \tag{1.31}$$

For the time transformation function $\mu_i(t)$, it needs to be strictly monotonically increasing and confined on the support of t: $t_1 \leq \mu_1(t) < \mu_2(t) < ... < \mu_n(t) \leq t_n$. We assign multivariate Gaussian prior on the time transformation coefficient ϕ_i:

$$\boldsymbol{\phi}_i \sim N(\boldsymbol{\Upsilon}, \boldsymbol{\Sigma}_\phi) \tag{1.32}$$

where $\boldsymbol{\Upsilon}$ is coefficient for the identity time transformation function which satisfies $\mu(t; \boldsymbol{\Upsilon}) = t$. For the coefficient of shape function $\boldsymbol{\beta}$, we also assign multivariate Gaussian prior on it:

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\beta) \tag{1.33}$$

The estimation of $\boldsymbol{\phi}$ and $\boldsymbol{\beta}$ is actually a curve-fitting problem. Hence the smoothness of $\mu_i(\cdot)$ and $m(\cdot)$ needs to be controlled. It is achieved through the use of bayesian P-splines proposed by [Lang and Brezger, 2004]. This method places a first order random walk shrinkage prior on the coefficient parameter:

$$\beta_k = \beta_{k-1} + e_k, \quad e_k \sim N(0, \lambda)$$

We assume that $\beta_0 = 0$. It can be shown that the precision matrix of priors on $\boldsymbol{\beta}$ is $\boldsymbol{\Omega}/\lambda$, where $\boldsymbol{\Omega}$

is in a special banded pattern:

$$\mathbf{\Omega} = \begin{pmatrix} 2 & -1 & 0 & & & 0 \\ -1 & 2 & -1 & \ddots & & \\ 0 & -1 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & -1 & 0 \\ & & \ddots & -1 & 2 & -1 \\ 0 & & & 0 & -1 & 1 \end{pmatrix}$$

A similar shrinkage prior is placed on $\boldsymbol{\phi}_i$

$$\phi_{ik} - \Upsilon_k = (\phi_{i(k-1)} - \Upsilon_{k-1}) + \eta_k, \quad \eta_k \sim N(0, \sigma_\phi^2) \tag{1.34}$$

Assuming $\phi_{i0} - \Upsilon_0 = 0$ so that $\boldsymbol{\phi}_i \sim N(\boldsymbol{\Upsilon}; \sigma_\phi^2 \boldsymbol{P}^{-1})$. $\boldsymbol{P}$ has the same pattern as $\mathbf{\Omega}$ but with different dimension. Q is the length of the vector parameter $\boldsymbol{\phi}_i$. Time transformation function is monotonic increasing. It is realized by the constraint on components of $\boldsymbol{\phi}_i$: $t_1 = \phi_{i1} < \phi_{i2} < ... < \phi_{iQ} = t_n$. This is because according to [de Boor, 1978]:

$$\mu_i'(t) = \frac{1}{h} \sum (\phi_{ik+1} - \phi_{ik}) B_k(t, q-1) \tag{1.35}$$

Therefore $\phi_{ik+1} > \phi_{ik}$ can result in monotonic increasing function $\mu_i(t)$.

Stage Three. We also assign priors on variances and hyperparameters:

$$\begin{aligned} a_0 &\sim N(m_a; \sigma_{a0}^2) \\ c_0 &\sim N(m_c; \sigma_{c0}^2) \\ 1/\sigma_a^2 &\sim Gamma(a_a, b_a) \\ 1/\sigma_c^2 &\sim Gamma(a_c, b_c) \\ 1/\sigma_\epsilon^2 &\sim Gamma(a_\epsilon, b_\epsilon) \\ 1/\lambda &\sim Gamma(a_\lambda, b_\lambda) \\ 1/\sigma_\phi^2 &\sim Gamma(a_\phi, b_\phi) \end{aligned} \tag{1.36}$$

The restriction of fixed starting and termination time point for $\boldsymbol{\phi}_i(t)$ brings unsatisfactory registration result. In reality, different individuals start and end at different phases. To make the regis-

tration accomodated to such situation, we use a quantity Δ to relax the starting and end points. So now the time interval is $T = [t_1 - \Delta, t_n + \Delta]$.

1.2.3.2 Posterior Inference via MCMC

Let $\boldsymbol{\theta} = (\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{\beta}', \boldsymbol{\phi}', c_0, a_0, \sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \sigma_\phi^2, \lambda)$ denote the vector of all parameters, then the posterior distribution is:

$$\begin{aligned} f(\boldsymbol{\theta}|\boldsymbol{Y}) &= f(\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{\beta}', \boldsymbol{\phi}', c_0, a_0, \sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \sigma_\phi^2, \lambda|\boldsymbol{Y}) \\ &\propto f(\boldsymbol{Y}|\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{\beta}', \boldsymbol{\phi}', \sigma_\epsilon^2) f(\boldsymbol{c}, \boldsymbol{a}|c_0, a_0, \sigma_c^2, \sigma_a^2) \\ &\quad \times f(\boldsymbol{\beta}|\lambda) f(\boldsymbol{\phi}|\sigma_\phi^2) f(c_0|\sigma_{c_0}^2) f(a_0|\sigma_{a_0}^2) \\ &\quad \times f(\sigma_\epsilon^2|a_\epsilon, b_\epsilon) f(\sigma_c^2|a_c, b_c) f(\sigma_a^2|a_a, b_a) \\ &\quad \times f(\lambda|a_\lambda, b_\lambda) f(\sigma_\phi^2|a_\phi, b_\phi) \end{aligned} \tag{1.37}$$

Since the joint distribution is intractable to directly sample from, then MCMC sampling method is applied to draw samples from it. Except $\boldsymbol{\phi}_i$, all the other parameters have closed-form full conditional distribution since they have conjugate prior. For $\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{\beta}', \boldsymbol{\phi}', c_0, a_0$, their full conditional distribution is Gaussian distribution. For $\sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \sigma_\phi^2, \lambda$, their full conditional distribution is inverse Gamma. So Gibbs sampler is suitable to draw samples from these distributions. For $\boldsymbol{\phi}_i$, the mean function of $y_i(t)$ is $c_i + a_i \mathbf{B}'_m(\mathbf{B}'_\mu(t)\boldsymbol{\phi}_i)\boldsymbol{\beta}$, which is nonlinear in $\boldsymbol{\phi}_i$. So it is hard to find a closed-form posterior distribution for $\boldsymbol{\phi}_i$. The Metropolis-Hasting algorithm is adopted to simulate samples from the posterior of $\boldsymbol{\phi}_i$. The proposal density is calibrated so that the acceptance rate for Metropolis-Hasting algorithm is between 0.25 to 0.75. In summary, this MCMC sampling method is a mixture of Gibbs and Metropolis-Hasting algorithm. The algorithm is as below:

Let $\boldsymbol{t} = (t_1, ..., t_n)'$ denote the time vector, $\boldsymbol{Y} = (y_1(t_1), ..., y_1(t_n), ..., y_N(t_1), ..., y_N(t_n))$ denote the vector of all the observations for one subject.

1. Update mean shape function parameter $\boldsymbol{\beta}$ by:

$$(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\theta}_{-\beta}) \sim N(\boldsymbol{m}_\beta; \boldsymbol{V}_\beta) \tag{1.38}$$

where $\boldsymbol{V}_{\beta}^{-1} = \Sigma_{\beta}^{-1} + 1/\sigma_{\epsilon}^2 \boldsymbol{X}'\boldsymbol{X}$, $\boldsymbol{m}_{\beta} = \boldsymbol{V}_{\beta}[1/\sigma_{\epsilon}^2 \boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{C})]$, $\boldsymbol{C} = (c_1\mathbf{1}', ..., c_N\mathbf{1}')'$, and $\boldsymbol{X} = (a_1\mathbf{B}_m(\mu(\boldsymbol{t};\boldsymbol{\phi}_1))', ..., a_N\mathbf{B}_m(\mu(\boldsymbol{t};\boldsymbol{\phi}_N))')'$.

2. For $i = 1, ..., N$, update (c_i, a_i) by:

$$(c_i, a_i|\boldsymbol{Y}, \boldsymbol{\theta}_{-(c_i,a_i)}) \sim N(\boldsymbol{m}_l; \Sigma_l) \tag{1.39}$$

where $\Sigma_l^{-1} = [\Sigma_{c,a}^{-1} + 1/\sigma_{\epsilon}^2 \boldsymbol{W}'\boldsymbol{W}]$, $\boldsymbol{m}_l = \Sigma_l \times [\Sigma_{c,a}^{-1} \times (c_0, a_0)' + 1/\sigma_{\epsilon}^2 \times \boldsymbol{W}'\boldsymbol{Y}_i]$, $\boldsymbol{Y}_i = (y_i(t_1), ..., y_i(t_n))'$, $\Sigma_{c,a} = diag(\sigma_c^2, \sigma_a^2)$, $\boldsymbol{W} = [\mathbf{1}, \mathbf{B}_m(\mu(\boldsymbol{t};\boldsymbol{\phi}_i))\boldsymbol{\beta}]$.

3. Update the error variance parameter σ_{ϵ}^2 by:

$$(1/\sigma_{\epsilon}^2|\boldsymbol{Y}, \boldsymbol{\theta}_{-\sigma_{\epsilon}^2}) \sim Gamma(a_{\epsilon}^*, b_{\epsilon}^*) \tag{1.40}$$

where $a_{\epsilon}^* = a_{\epsilon} + nN/2$, $b_{\epsilon}^* = b_{\epsilon} + 1/2\Sigma_{i=1}^{N}(\boldsymbol{Y}_i - \tilde{\boldsymbol{m}}_i)'(\boldsymbol{Y}_i - \tilde{\boldsymbol{m}}_i)$, $\tilde{\boldsymbol{m}}_i = c_i\mathbf{1} + a_i\mathbf{B}_m(\mu(\boldsymbol{t};\boldsymbol{\phi}_i))\boldsymbol{\beta}$.

4. For $i = 1, ..., N$, update $\boldsymbol{\phi}_i$ by;

   (a) For $j = 1, ..., Q$:

      i. propose ϕ_{ij}^* from its support.

      ii. Calculate the posterior ratio $r = \frac{L(\boldsymbol{Y}_i|\boldsymbol{\phi}_i^*, \boldsymbol{\theta}_{-\phi_i})p(\boldsymbol{\phi}_i^*)}{L(\boldsymbol{Y}_i|\boldsymbol{\phi}_i, \boldsymbol{\theta}_{-\phi_i})p(\boldsymbol{\phi}_i)}$

      iii. Accept $\boldsymbol{\phi}_{ij}^*$ with probability $min(1, r)$.

5. Update hyperparameters $c_0, a_0, \sigma_c^2, \sigma_a^2, \lambda, \sigma_{\phi}^2$.

After M draws of samples from posterior, the mean shape function is calculated by:

$$m(t) = \bar{c}_0 + \bar{a}_0\mathbf{B}_m'(t)\bar{\boldsymbol{\beta}} \tag{1.41}$$

where $\bar{c}_0$, $\bar{a}_0$ and $\bar{\boldsymbol{\beta}}$ are the averages over the M samples.

[Telesca and Inoue, 2008] compared BHCR with the landmark method and self-modeling warping function method by [Gervini and Gasser, 2004]. From the results of simulation study and case study, BHCR consistently shows better performance then those two methods.

In a conclusion, the Bayesian hierarchical curve registration offers a complete Bayesian framework for curve alignment. Unlike most other curve registration methods, it does not require presmoothing of data. Furthermore, this method makes it straightforward to draw inference on the

estimated parameters. The simulation and case study shows it has satisfactory or even better performance than existing methods.

1.3 Functional Regression

1.3.1 Functional Principle Component Analysis

Before we jump into reviewing different types of functional regression, there is one key method worth discussion first. This method is functional principle component analysis (FPCA). It helps to find features in data by characterizing the "typical" functions and presents the covariance structure in the data. It decomposes the variation in functional data onto different directions using weighting functions. By examining the weighting function, people are able to find out how variations are distributed along the independent variable range (e.g., time). The monograph of [Ramsay and Silverman, 2005] provides a comprehensive review of functional PCA.

The basic idea of FPCA is to decompose covariance function of functional data $\boldsymbol{X}$ into orthonormal basis functions. Such decomposition exists guaranteed by Mercer's lemma, and realized by Karhunen-Loève expansion. Mercer's lemma states that assume the covariance function K is continuous square-integrable, then there exists an orthonormal sequence ϕ_i of continuous function and a non-increasing sequence ξ_i of positive numbers such that:

$$K(u,v) = \sum_{i}^{\infty} \xi_i \phi_i(u) \phi_i(v),\ u, v \in \mathbb{R} \tag{1.42}$$

Karhunen-Loève expansion is defined as

$$X(u) = \mu_x(u) + \sum_{i}^{\infty} \sqrt{\xi_i} \zeta_i \phi_i(u) \tag{1.43}$$

where $\{\zeta_i\}$ are uncorrelated random variables. ξ_i and $\phi_i(u)$ are eigenvalues and eigenfunctions respectively of the covariance function. By Karhunen-Loève expansion, FPCA is able to obtain an approximation of functional data $X(u)$ by:

$$\widehat{X(u)} = \mu_x(u) + \sum_{i}^{M} \sqrt{\xi_i} \zeta_i \phi_i(u) \tag{1.44}$$

The eigenvalue ξ_i can be interpreted as a measure of variation in X on the ϕ_i direction. In practice, when people choose number of eigenfunctions M, at least 85% variation should be counted by the first M eigenfunctions.

Functional PCA is an important way to explore functional data. However, it lacks the ability of using functional data to explain outcome responses. The way to achieve this goal is through functional regression. Functional regression is the area in functional data analysis that has received the most development in methodology and application. Depending on the form of response variable and predictor variable, functional regression can be categorized into three groups: scalar-on-function (functional predictor regression), function-on-scalar (functional response regression) and function-on-function regression. Among these three types of functional regression, the functional predictor regression is the most widely used one, which we will introduce first.

1.3.2 Functional Predictor Regression

The idea of using functional predictor to predict scalar response originated from [Hastie and Mallows, 1993]. The author noticed that in chemometrics study, people used discretized functions or signals to predict outcome as below:

$$\begin{aligned} E(Y_i) &= \int X_i(t)\beta(t)dt \\ &\approx \sum_j^p X_{ij}\beta_j \end{aligned} \tag{1.45}$$

where $X_i(t)$ is the functional predictor and $\beta(t)$ is the functional coefficient. There are two major problems with such method. It discarded the functional nature of $x(t)$ and disregarded the fact that X_{ij}'s are in spatial or time order. The other problem is that the number of predictors is much larger than the number of outcome observations, and predictors are closely related. The fitted model has poor predictive ability and interpretability. A more appropriate solution is to retain the functional nature of $x(t)$ and smooth it, which involves expressing $x(t)$ in terms of a linear combination of basis functions. Accordingly, the coefficient $\beta(t)$ is also treated as function and smoothed by basis functions. The regression model is denoted by the following formula:

$$Y_i = \boldsymbol{H}_i'\boldsymbol{\gamma} + \int_T X_i(t)\beta(t)dt + \epsilon_i, \quad i = 1, ..., N \tag{1.46}$$

with both scalar predictor $\boldsymbol{H}_i$ and functional predictor $X_i(t)$. To interpret such model, the functional coefficient $\beta(t)$ is basically re-weighting $X_i(t)$. This re-weighting is similar to the discrete weights β_j in 1.45 except it is a smoother transition in the weighting scheme. Notice that $X_i(t)$ is a subject-specific predictor, and $\beta(t)$ is a population level function. Hence the weight close to zero denotes a weak relation between the subject-level area and the outcome, and large weight denotes a strong relation between the subject-level area and the outcome.

Functional regression with scalar response is a fast growing area and has many important applications. A sample of papers in this field includes [Ferraty and Vieu, 2002], [James, 2002], [Müller and Stadtmüller, 2005], [James and Silverman, 2005], [James et al., 2009], [Goldsmith et al., 2011]. To estimate $\beta(t)$, a common strategy is to express $X_i(t)$ and $\beta(t)$ separately by linear combination of basis functions. Let $X_i(t) = \boldsymbol{c}_i'\boldsymbol{\phi}(t)$, $\beta(t) = \boldsymbol{\psi}(t)'\boldsymbol{b}$, $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ are vector of basis functions. The integral can be expressed as:

$$\begin{aligned}\int_T X_i(t)\beta(t)dt &= \int_T \boldsymbol{c}_i'\boldsymbol{\phi}(t)\boldsymbol{\psi}(t)'\boldsymbol{b}dt \\ &= \boldsymbol{c}_i'\boldsymbol{J}_{\phi\psi}\boldsymbol{b}\end{aligned}$$

where $\boldsymbol{J}_{\phi\psi} = \int_T \boldsymbol{\phi}(t)\boldsymbol{\psi}(t)'dt$. In practice, $\boldsymbol{c}_i$ is usually calculated in pre-smooth step of $X_i(t)$. The challenge is to control the smoothness of the functional predictor $\beta(t)$ and make the model more interpretable. The smoothness can be controlled by the number of basis functions. When using b-spline basis, it is controlled through the location and number of knots. An easy-to-implement strategy is to include a large number of basis functions, and penalize the roughness of functional coefficient $\beta(t)$. For example, in [Ramsay and Silverman, 2005] the penalized residual sum of squares to minimize is defined as:

$$PENSSE = \sum_{i=1}^{N}[Y_i - \boldsymbol{H}_i'\boldsymbol{\gamma} - \int_T X_i(t)\beta(t)dt]^2 + \lambda\int[L\beta(t)]^2dt \tag{1.47}$$

where L is a linear differential operator, and λ is the smoothing parameter, which can be determined by cross-validation. The method simplifies the modeling part by pre-smoothing functional data $X(t)$. [Goldsmith et al., 2011] introduced the penalized functional regression which estimates both $X(t)$ and $\beta(t)$. It considers the functional data $X(t)$ as a stochastic process with observation error, which makes this method more realistic. We will discuss this method in detail in section 5.3.

In some applications each subject can be visited repeatedly, which brings the challenge of analyzing longitudinal functional data. This area is quite new and limited number of methods is available. One idea to untangle the problem through longitudinal functional principal component analysis (LFPCA), proposed by [Di et al., 2009] and [Greven et al., 2010]. This method models the longitudinal functional data by a two-way ANOVA model, which involves overall mean function, visit-specific shift, subject-specific shift, and residual visit- and subject-specific shift. The first two are treated as fixed effect. The latter two are modeled as stochastic process. This method is an extension to the functional PCA, and it doesn't involve using functional predictors to predict the scalar response y_i. Another available method is longitudinal penalized functional regression proposed by [Goldsmith et al., 2012]. This method is a natural extension to model [1.46] by making the design matrix of coefficient function $\beta(t)$ consist of random effects.

Different from the scenario described above, another kind of repeatedly measured functional data are: each subject has a series of curves with similar morphological feature, and each subject has one single outcome. The motivating data come from the case that in intensive-care unit (ICU) intracranial pressure (ICP) is monitored for patients who are suffering severe brain damage. ICP is a critical measurement for diagnosing and managing these neurosurgical patients. Each ICP pulse is a curve and can be treated as a functional predictor. It is measured repeatedly over time (each patient has about 200 ICP pulses). Our aim is to explore the relationship between ICP pulses and clinical outcome (e.g., living status). Given the large number of repeated measured curves for each subject, it is impossible to include them all in the regression model. Hence it is necessary to generate a mean curve function based on all curves available for each subject. A naive way to achieve this goal is to average over all curves. However, ignoring the time variability from curve to curve will cause the cross-sectional mean curve to be over-smoothed and missing certain features. For instance, local peak and valley of each curve occur at different time points, and then the cross-sectional mean curve may have a flat segment instead of the peak and valley. A more appropriate way to solve this problem is through curve registration, which synchronizes curves by time warping function. After the curves are aligned, an estimated mean function is calculated and used in functional regression as predictor. In addition to the mean function, we also include the variance of amplitude and phase as scalar predictor in our regression model to account for the

variability among curves.

There are several methods available for curve registration. For example, landmark registration by [Gasser and Kneip, 1995] identifies landmarks of curve and align all curves according to the landmark. Self-modeling registration by [Gervini and Gasser, 2004] introduces a curve-specific unknown time transformation function and approximates it by basis functions. In this paper, we use the Bayesian Hierarchical Curve Registration method (BHCR) proposed by [Telesca and Inoue, 2008], which has similar idea of time transformation function as the self-modeling registration, but conducted under bayesian framework.

1.3.2.1 Stochastic Process and Karhunen-Loève Expansion

Before introducing the penalized functional regression by [Goldsmith et al., 2011], we will first give a brief introduction to stochastic process and Karhunen-Loève Expansion used in such case. Stochastic process models a collection of random variables evolving over time. It is probabilistic counterpart to deterministic process. The variability or randomness are time-dependent. A stochastic process can be considered as a function of random outcomes and observed time parameter t, and it is denoted by $\{X(t), t \in T\}$ or simply $X(t)$. The mean function of a stochastic process is defined by:

$$\mu_X(t) = \int X(t)dt$$

and the covariance function is defined by:

$$\text{Cov}(X(t_i), X(t_j)) = \int (X(t_i) - \mu_X(t_i))((X(t_j) - \mu_X(t_j))dt$$

Karhunen-Loève expansion is a representation of a stochastic process as an infinite linear combination of orthonormal functions. It is an advanced mathematical algorithm to achieve both noise filtering and data compression in processing signals.

The expansion of a deterministic periodic signal $x(t)$ into a basis of orthonormal functions is typified by the classical Fourier series:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}[a_n\text{cos}(nt) + b_n\text{sin}(nt)], \quad (-\pi \leq t \leq \pi)$$

It is natural to extend this deterministic case into probabilistic case, so called stochastic process. Let $X(t)$ denote a stochastic process, the starting formula to expand this process is

$$X(t) = \sum_{n=1}^{\infty} Z_n \phi_n(t) \tag{1.48}$$

$\phi_n(t)$ denotes orthonormal function. It satisfies

$$\int_0^T \phi_m(t)\phi_n(t)dt = \delta_{mn} \tag{1.49}$$

where the δ_{mn} is the Kronecker symbol, defined by $\delta_{mn} = 0$ for $m \neq n$ and $\delta_{nn} = 1$. Recall that in Fourier series, the coefficient a_n and b_n are computed through:

$$a_n = \frac{1}{n}\int x(t)\text{cos}(nt)dt, \;\; b_n = \frac{1}{n}\int x(t)\text{sin}(nt)dt$$

For Karhunen-Loève expansion, the coefficient Z_n is computed in a similar way. Considering the randomness of $X(t)$, its behavior is made up of two parts: the deterministic part represented by $\phi_n(t)$ changing in time and the random part represented by Z_n, which is a random variable (not stochastic process). By doing this, the KLT separates the probabilistic behavior of the random function from its behavior in time. Similar as Fourier series, Z_n is obtained through:

$$Z_n = \int X(t)\phi_n(t)dt \tag{1.50}$$

This equation means the random variable Z_n is obtained by projecting the stochastic process $X(t)$ over the corresponding eigenvector $\phi_n(t)$.

Without loss of generality, we can assume the mean function of the stochastic process is zero:

$$\sum_{n=1}^{\infty} \text{E}(Z_n)\phi_n(t) = 0 \tag{1.51}$$

Thus the random variable Z_n must have mean 0 too, which leads to an equation for its variance:

$$\sigma_{Z_n}^2 = \text{E}(Z_n)$$

We introduce a sequence of positive numbers λ_n such that each λ_n is the variance of the corresponding Z_n. It can be proved that the orthonormal function $\phi_n(t)$ is the eigenfunction of the

correlation $\mathrm{E}(X(t_1)X(t_2))$, and λ_n is the corresponding eigenvalue. That means once the correlation function for the stochastic process is known, the process can be represented by an infinite linear combination of orthonormal basis. Another way to understand this property is once the mean and covariance function of a stochastic process are determined, it can be expanded by KL expansion.

1.3.2.2 Penalized Functional Regression

[Goldsmith et al., 2011] introduced the penalized functional regression method based on the following model:

$$\begin{aligned} Y_i &\sim \mathrm{EF}(\mu_i, \eta) \\ g(\mu_i) &= \alpha + \int X_i(s)\beta(s)ds + \boldsymbol{Z}_i\boldsymbol{\gamma} \end{aligned} \tag{1.52}$$

Here $\mathrm{EF}(\mu_i, \eta)$ denotes an exponential family distribution with mean μ_i and dispersion parameter η. $g(\cdot)$ is the link funciton. This penalized functional regression is designed for the assumption that the functional predictor $X_i(t)$ is often measured with error. $W_i(t)$ is used to denote the actual observed functional predictor, $W_i(t) = X_i(t) + \epsilon_i(t)$, where $\epsilon_i(t)$ is a mean-zero white noise process with variance σ_ϵ^2. Thus, for subject i the data typically are of the form $[Y_i, \{W_i(t_{ij}) : t_{ij} \in [0,1]\}, Z_i]$. Of interest is the function $\beta(t)$,which characterizes the relationship between the transformed mean of $\boldsymbol{Y}$ and the covariate of interest $X(\cdot)$. Both $X_i(t)$ and $\beta(t)$ can be expanded by linear combination of basis functions. The number of components is chosen to be large, and the smoothness is controlled by smoothing parameter. The estimation process contains two stages: the estimation of $X_i(t)$ and the estimation of $\beta(t)$.

Estimation of $X_i(t)$ Consider the functional predictor $X_i(t)$ as a stochastic process, it can be expanded into orthonormal basis obtained from its covariance operator $K^W(s,t)$:

$$X_i(t) = \sum_{j=1}^{K_x} c_{ik}\phi_k(t) \tag{1.53}$$

where $\boldsymbol{\phi} = \{\phi_1(t), ..., \phi_{K_x}(t)\}$ is the collection of the first K_x eigenfunctions of the covariance matrix $K^X(s,t) = \mathrm{Cov}\{X_i(s), X_i(t)\}$. Assuming $X_i(t)$ is observed with error: $W_i(t) = X_i(t) +$

$\epsilon_i(t)$ where $\epsilon_i(t)$ is a mean-zero white noise process with variance σ_ϵ^2. For the observed data $W_i(t)$, the covariance operator is:

$$K^W(s,t) = K^X(s,t) + \sigma_\epsilon^2 \delta_{ts} \tag{1.54}$$

where $K^W(s,t) = \text{Cov}\{W_i(s), W_i(t)\}$ is the covariance operator on the observed functions. $\delta_{ts} = 1$ if $t = s$ and is 0 otherwise. To estimate $K^X(s,t)$, a moment estimator of $\hat{K}^W(s,t)$ is constructed from observed data. Then the estimator is smoothed for $s \neq t$ by the method introduced in [Staniswalis and Lee, 1998] and [Yao et al., 2003]. Then with estimated $\hat{K}^X(s,t)$, $X_i(t)$ is expanded by truncated Karhunen-Loéve expansion where $c_{ik} = \int_0^1 X_i(t)\phi_k(t)dt$. Unbiased estimator of c_{ik} is obtained by Riemann sum approximation to the integral $\int W_i(t)\phi_k(t)dt$. This method works well when data are densely sampled. When it is not the case, a better alternative is to obtain the best linear unbiased predictor (BLUP) or posterior modes in the following mixed effects model:

$$\begin{aligned} W_i(t) &= \textstyle\sum_{k=1}^{K_x} c_{ik}\phi_k(t) + \epsilon_i, \\ c_{ik} &\sim N(0, \sigma_c^2), \quad \epsilon_i \sim N(0, \sigma_\epsilon^2) \end{aligned} \tag{1.55}$$

$X_i(t)$ is estimated by plugging in estimated c_{ik} into 1.53.

Estimation of $\beta(t)$ $\beta(t)$ is expanded by spline basis $\boldsymbol{\psi}(t) = \{\psi_1(t), ..., \psi_{K_b}(t)\}$:

$$\beta(t) \approx \sum_{k=1}^{K_b} b_k \psi_k(t) \tag{1.56}$$

The integral becomes:

$$\int X_i(t)\beta(t) = \int \boldsymbol{c}_i' \boldsymbol{\phi}'(t)\boldsymbol{\psi}(t)dt = \boldsymbol{c}_i' \boldsymbol{J}_{\phi\psi} \boldsymbol{b} \tag{1.57}$$

The basis coefficient vector $\boldsymbol{b}$ needs to be estimated. To control for smoothness, a penalty matrix $\boldsymbol{P}$ for $\boldsymbol{b}$ is introduced. Then the model can be reformulated as:

$$\begin{aligned} \boldsymbol{Y}|\boldsymbol{X}(\boldsymbol{t}) &\sim \text{EF}(\boldsymbol{\mu}, \boldsymbol{\gamma}) \\ g(\boldsymbol{\mu}) &= [\mathbf{1}\ \boldsymbol{C}\boldsymbol{J}_{\phi\psi}\ \boldsymbol{Z}][\alpha\ \boldsymbol{b}\ \boldsymbol{\gamma}]^T \\ \boldsymbol{b} &\sim N(\mathbf{0}, \boldsymbol{P}) \end{aligned} \tag{1.58}$$

This model is a mixed effects model. So the parameter estimation can be obtained by typical mixed effects model estimation methods. Typical inferential technique can also produce variance-covariance estimates, which leads to confidence interval for estimate of $\beta(t)$. The number of basis functions used in truncated KL expansion and spline expansion is K_x and K_b respectively. They are tuning parameters and considered to be important in practice. In penalized functional regression, they are chosen to be large. K_b is set to be 35, and K_x must be larger than K_b because of identifiability constraint. Penalized functional regression is implemented in R package "refund". [Goldsmith et al., 2011] conducted simulation study which shows this package is computationally fast.

Longitudinal penalized functional regression [Goldsmith et al., 2012] introduced the longitudinal penalized functional regression model to fit data when the functional predictor and scalar outcome are repeatedly measured. In that paper the regression formula is based on model with multiple functional predictors. Here for simplicity and consistency we only consider model with one functional predictor, and it is easy to generalize to multiple functional predictor scenario.

$$\begin{aligned} \boldsymbol{Y}_{ij} &\sim \text{EF}(\mu_{ij}, \eta) \\ g(\mu_{ij}) &= \int X_{ij}(s)\beta(s)ds + \boldsymbol{Z}_{ij}\boldsymbol{\gamma} + \boldsymbol{W}_{ij}\boldsymbol{b}_i \end{aligned} \tag{1.59}$$

Here $\boldsymbol{Y}_{ij}$ is the functional observation of the i's subject at visit j, $\boldsymbol{Z}_{ij}\boldsymbol{\gamma}$ is the fixed effect component, $W_{ij}\boldsymbol{b}_i$ is the random effect component and $\int X_{ij}(s)\beta(s)ds$ is the functional effect. $\boldsymbol{b}_i \sim N(\mathbf{0}, \sigma_b^2\mathbf{I})$ is subject-specific random effect coefficient, which is used to account for the correlation between repeated observations at the subject level. $\beta(s)$ and $\boldsymbol{\gamma}$ are population level coefficient and do not vary across visits. The estimation of the functional component is done by two steps similar as in penalized functional regression:

1. Express the functional predictor X_{ij} by a large number of functional principle components obtained from a smooth estimator of the covariance matrix

2. Express the functional coefficient by penalized splines.

These two steps are quite similar as in 1.3.2.2. The functional effect component can be finally expressed as:

$$\int X_{ij}(s)\beta(s)ds = \boldsymbol{c}_{ij}'\boldsymbol{J}_{\phi\psi}\boldsymbol{g} \tag{1.60}$$

The regression model becomes:

$$\begin{aligned} \boldsymbol{Y}|\boldsymbol{X}(\boldsymbol{t}) &\sim \text{EF}(\boldsymbol{\mu}, \boldsymbol{\gamma}, \boldsymbol{u}) \\ g(\boldsymbol{\mu}) &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} \end{aligned} \tag{1.61}$$

$$\boldsymbol{u} \sim N\left\{ \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \sigma_b^2\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sigma_{g_1}^2\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sigma_{g_K}^2\mathbf{I} \end{pmatrix} \right\}$$

$\boldsymbol{X}$ are $\boldsymbol{Z}$ are new matrices different from the previous ones. $\boldsymbol{X}$is the design matrix consisting of scalar covariates and fixed effects. $\boldsymbol{Z}$ is the design matrix consisting of subject-specific random effects and random effects in the modeling of the functional coefficient. In sum, similar as in 1.3.2.2, the model estimation can be done using standard mixed effects model technique. Both functional coefficient and random effects can be estimated. This method is also implemented in the R package "refund".

1.3.2.3 Bayesian Functional Predictor Regression

In this section we will introduce Bayesian framework for functional predictor regression. The scalar response can be continuous or binary, and the functional predictor is formulated by B-spline. Two case studies will also be demonstrated.

Continuous Scalar Response Let Y_i denote the ith continuous response variable of the ith subject, $\boldsymbol{H}$ denote the design matrix of scalar predictor. Assuming for each subject there is one functional predictor $m_i(t)$ associated with it. The functional predictor is assessed over time t but it is easy to generalize over other continuum. There is also functional coefficient $\boldsymbol{\nu}(t)$ associated with the functional predictor. The complete functional regression model is as below:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\gamma} + \int \boldsymbol{M}(t)\boldsymbol{\nu}(t)dt + \boldsymbol{\epsilon} \tag{1.62}$$

where each row of $\boldsymbol{M}$ is the functional predictor $m_i(t)$ for subject i. $m_i(t)$ is expressed by the following formula the same as in Chapter 3:

$$m_i(t) = c_i + a_i m(\mu_i(t); \boldsymbol{\beta}) = c_i + a_i \mathbf{B}'_m(\mu_i(t))\boldsymbol{\beta} \tag{1.63}$$

$\mu_i(t)$ is time transformation function, which can be expanded by $\mu_i(t) = \mathbf{B}'_\mu(t)\boldsymbol{\phi}_i$. $\mathbf{B}_m(t)$ and $\mathbf{B}_\mu(t)$ are vectors of B-spline basis functions. $m_i(t)$ is estimated through the Bayesian curve registration process. Notice that $m_i(t)$ is an estimator from curve registration, not data, to account for this, we also add the estimated variances of scale parameters and phase parameter into the model, which are σ_a^2, σ_c^2, and σ_ϕ^2. These three variances are in the fixed effect.

The functional coefficient $\boldsymbol{\nu}(t)$ can also be expanded by B-spline basis functions: $\boldsymbol{\nu}(t) = \mathbf{B}_\alpha(t)\boldsymbol{\alpha}$. So the model becomes:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\gamma} + \boldsymbol{M}\mathbf{B}_\alpha\boldsymbol{\alpha} + \boldsymbol{\epsilon} \tag{1.64}$$

where $\boldsymbol{\epsilon}$ has Gaussian distribution $N(\mathbf{0}, \sigma_\epsilon^2\mathbf{I})$.

We propose a two-stage Bayesian functional regression method. In the first stage, the Bayesian hierarchical curve registration is performed to obtain estimation of $m_i(t)$, σ_a^2, σ_c^2, and σ_ϕ^2. In the second stage, the estimated predictors are plugged into (1.64) and the regression model parameters are estimated by MCMC sampling. We assign Gaussian prior on $\boldsymbol{\gamma}$, and use the first order random walk on $\boldsymbol{\alpha}$ to control the smoothness.

$$\begin{aligned} \boldsymbol{\gamma} &\sim N(\boldsymbol{\gamma}_0, \lambda_\gamma\mathbf{I}) \\ \boldsymbol{\alpha} &\sim N(\boldsymbol{\alpha}_0, \lambda_\alpha\boldsymbol{\Omega}^{-1}) \end{aligned} \tag{1.65}$$

So the full conditional posterior distribution of $\boldsymbol{\gamma}$ is:

$$\boldsymbol{\gamma}|\boldsymbol{Y}, \boldsymbol{\gamma}_0, \sigma_\epsilon^2, \boldsymbol{\alpha}, \lambda_\gamma, \lambda_\alpha \sim N(\boldsymbol{m}_\gamma, \boldsymbol{V}_\gamma)$$

$$\boldsymbol{m}_\gamma = \boldsymbol{V}_\gamma[\boldsymbol{H}'(\boldsymbol{Y} - \boldsymbol{M}\mathbf{B}\boldsymbol{\alpha})/\sigma_\epsilon^2 + \boldsymbol{\gamma}_0/\lambda_\gamma)], \boldsymbol{V}_\gamma = (1/\lambda_\gamma + 1/\sigma_\epsilon^2)^{-1}(\boldsymbol{H}'\boldsymbol{H} + \mathbf{I})^{-1} \tag{1.66}$$

The full conditional posterior distribution of $\boldsymbol{\alpha}$ is:

$$\boldsymbol{\alpha}|\boldsymbol{Y}, \sigma_\epsilon^2, \boldsymbol{\gamma}, \lambda_\gamma, \lambda_\alpha \sim N(\boldsymbol{m}_\alpha, \boldsymbol{V}_\alpha)$$

$$\boldsymbol{m}_\alpha = \boldsymbol{V}_\alpha(\boldsymbol{M}\mathbf{B})'(\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\gamma})/\sigma_\epsilon^2, \boldsymbol{V}_\alpha = (1/\lambda_\alpha\boldsymbol{\Omega} + 1/\sigma_\epsilon^2(\boldsymbol{M}\mathbf{B})'(\boldsymbol{M}\mathbf{B}))^{-1} \tag{1.67}$$

We assign gamma distribution on σ_ϵ^2, λ_γ, and λ_α, normal distribution on $\boldsymbol{\gamma}_0$. So their full conditionals are:

$$\begin{aligned}
\sigma_\epsilon^2|\boldsymbol{Y}, \boldsymbol{\gamma}, \boldsymbol{\alpha} &\sim IG(a_\epsilon + 1/2N, b_\epsilon + 1/2\textstyle\sum_{i=1}^N (Y_i - \boldsymbol{H}_i\boldsymbol{\gamma} - \boldsymbol{M}_i\mathbf{B}\boldsymbol{\alpha})^2) \\
\lambda_\gamma|\boldsymbol{Y}, \sigma_\epsilon^2, \boldsymbol{\gamma} &\sim IG(a_\gamma + 1/2Q_\gamma, b_\gamma + 1/2(\boldsymbol{\gamma} - \boldsymbol{\gamma}_0)'(\boldsymbol{\gamma} - \boldsymbol{\gamma}_0)) \\
\lambda_\alpha|\boldsymbol{Y}, \sigma_\epsilon^2, \boldsymbol{\alpha} &\sim IG(a_\alpha + 1/2Q_\alpha, b_\alpha + 1/2\boldsymbol{\alpha}'\boldsymbol{\Omega}\boldsymbol{\alpha}) \\
\boldsymbol{\gamma}_0|\boldsymbol{\gamma} &\sim N(\boldsymbol{m}_0, \boldsymbol{V}_{\gamma_0}),\ \boldsymbol{m}_0 = \boldsymbol{V}_{\gamma_0}\boldsymbol{\gamma}/\lambda_\gamma,\ \boldsymbol{V}_{\gamma_0} = (1/\lambda_\gamma + 1/\sigma_{\gamma 0}^2)^{-1}\mathbf{I}
\end{aligned} \tag{1.68}$$

where Q_γ and Q_α are the dimensions of $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$. Since these are all conjugate priors, Gibbs sampler can be used to sample from the full conditionals.

Binary Scalar Response For binary outcome, $Y_i \in \{0, 1\}$, assuming it has Bernoulli distribution, the functional model can be reformatted as:

$$\begin{aligned}
E(Y_i) &= \mu_i = g(\eta_i) \\
\eta_i &= \boldsymbol{H}_i\boldsymbol{\gamma} + \boldsymbol{M}_i\mathbf{B}\boldsymbol{\beta}
\end{aligned} \tag{1.69}$$

where $g(\cdot)$ is the link function. Let $\boldsymbol{\theta}$ denote the vector of all parameters, then the likelihood function becomes:

$$L(\boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{i=1}^N g(\eta_i)^{y_i}(1 - g(\eta_i))^{1-y_i} \tag{1.70}$$

We use the data augmentation method proposed by [Albert and Chib, 1993] to facilitate MCMC sampling. This data augmentation method uses the standard Gaussian cumulative density function $\Phi(\cdot)$ as the link function $g(\cdot)$. It also introduces independent latent random variables $Z_1, ..., Z_N$. Define $Y_i = 1$ if $Z_i > 0$ and $Y_i = 0$ if $Z_i \leq 0$. So $Pr(Y_i = 1) = Pr(Z_i > 0) = \Phi(\eta_i)$, and $Pr(Y_i = 0) = 1 - \Phi(\eta_i)$. Then the conditional posterior distribution of Z is a truncated Gaussian distribution as below:

$$\begin{aligned}
&Z_i|Y_i = 1, \boldsymbol{\theta} \sim N(\eta_i, 1) \quad \text{truncated at the left by } 0 \\
&Z_i|Y_i = 0, \boldsymbol{\theta} \sim N(\eta_i, 1) \quad \text{truncated at the right by } 0
\end{aligned} \tag{1.71}$$

The full conditional distributions of parameters other than $\boldsymbol{Z}$ are:

$$\boldsymbol{m}_\gamma = \boldsymbol{V}_\gamma[\boldsymbol{H}'(\boldsymbol{Z} - \boldsymbol{M}\mathbf{B}\boldsymbol{\alpha}) + \boldsymbol{\gamma}_0/\lambda_\gamma)], \boldsymbol{V}_\gamma = (1/\lambda_\gamma + 1)^{-1}(\boldsymbol{H}'\boldsymbol{H} + \mathbf{I})^{-1}$$

$$\boldsymbol{\alpha}|\boldsymbol{Z}, \boldsymbol{\gamma}, \lambda_\gamma, \lambda_\alpha \sim N(\boldsymbol{m}_\alpha, \boldsymbol{V}_\alpha)$$

$$\boldsymbol{m}_\alpha = \boldsymbol{V}_\alpha(\boldsymbol{M}\mathbf{B})'(\boldsymbol{Z} - \boldsymbol{H}\boldsymbol{\gamma}), \boldsymbol{V}_\alpha = (1/\lambda_\alpha\boldsymbol{\Omega} + (\boldsymbol{M}\mathbf{B})'(\boldsymbol{M}\mathbf{B}))^{-1}$$

$$\lambda_\gamma|\boldsymbol{\gamma} \sim IG(a_\gamma + 1/2Q_\gamma, b_\gamma + 1/2(\boldsymbol{\gamma} - \boldsymbol{\gamma}_0)'(\boldsymbol{\gamma} - \boldsymbol{\gamma}_0))$$

$$\lambda_\alpha|\boldsymbol{\alpha} \sim IG(a_\alpha + 1/2Q_\alpha, b_\alpha + 1/2\boldsymbol{\alpha}'\boldsymbol{\Omega}\boldsymbol{\alpha})$$

$$\boldsymbol{\gamma}_0|\boldsymbol{\gamma} \sim N(\boldsymbol{m}_0, \boldsymbol{V}_{\gamma_0}),\ \boldsymbol{m}_0 = \boldsymbol{V}_{\gamma_0}\boldsymbol{\gamma}/\lambda_\gamma,\ \boldsymbol{V}_{\gamma_0} = (1/\lambda_\gamma + 1/\sigma^2_{\gamma 0})^{-1}\mathbf{I} \tag{1.72}$$

The MCMC sampling algorithm is almost the same as when the response variable is continuous. One additional step is at the beginning of each iteration $\boldsymbol{Z}$ will be sampled from its conditional posterior on $\boldsymbol{Y}$.

Simulation Study

Continuous Scalar Response To investigate performance of the two-stage model, we generate 100 repeated data sets. Each data set is constructed in the same way: time grid is set on the interval $[0, 1]$ evenly divided by 100; we simulate 50 subjects, and each subject has 10 simulated curves in the form:

$$x_{ij}(t) \ = \ c_{ij} + a_{ij}m(\mu(t; \boldsymbol{\phi}_{ij})) + \epsilon_{ij},\ i = 1, ..., 50;\ j = 1, ..., 10$$

We simulate parameters from $c_{ij} \sim N(c_{0i}; \sigma^2_{c_i})$, $c_{0i} \sim N(0, 0.01)$, $\sigma^2_{c_i} \sim IG(20, 0.2)$, $a_{ij} \sim N(a_{0i}, \sigma^2_{a_i})I(a_j > 0)$, $a_{0i} \sim N(1, 0.01)$, $\sigma^2_{a_i} \sim IG(20, 0.2)$. Notice that c_{ij} and a_{ij} are subject-visit-specific parameter. $\sigma^2_{c_i}$ and $\sigma^2_{a_i}$ are subject-specific parameter. Curves of the same subject have the same mean shape function $m(t)$, and $m(t)$ is made to be different from subject to subject by s_i and u_i. s_i creates phase variation, and u_i creates amplitude variation.

$$m_i(t) = cos[\pi(t - s_i)] + sin[3\pi(t - s_i)] + u_i$$

$$s_i \sim \text{Uniform}(0.4, 0.6),\ u_i \sim N(0, 0.16)$$

The time transformation function is different among curves of the same subject, which reflects the time variation among the repeated measurements of the same function. $\mu_{ij}(t;\boldsymbol{\phi}_{ij})$ is the jth curve of the ith subject:

$$\mu_{ij}(t;\boldsymbol{\phi}_{ij}) = \mathbf{B}'(t)\boldsymbol{\phi}_{ij}$$

We use cubic b-spline with one internal knot at 0.5 to generate $\mu_{ij}(t;\boldsymbol{\phi}_{ij})$. The coefficient $\boldsymbol{\phi}_{ij}$ is generated from $N(\boldsymbol{\Upsilon}, \lambda_\phi \boldsymbol{\Omega}^{-1})$, $\lambda_\phi = 0.02$. For the functional coefficient $\boldsymbol{\alpha}(t)$, we simulated two different scenarios. Under scenario 1, the functional coefficient is continuous through the whole range of t, which means the relationship between response and functional predictor is changing gradually. Under scenario 2, the functional coefficient is continuous function in part of the variable range, and set to zero for the rest of the variable range, which means the response and functional predictor do not have any relationship in that range.

$$\alpha_1(t) = \cos(2\pi t)$$

$$\alpha_2(t) = \begin{cases} \sin(2\pi t) & \text{if } 0 \le t < 0.5 \\ 0 & \text{if } 0.5 \le t \le 1 \end{cases}$$

The response for the ith subject is generated by:

$$y_i = \sum_{s=1}^{100} m_i(t_s)\alpha(t_s)/100 + e_i$$

where e_i is generated from the Gaussian distribution $N(0, \sigma_e^2)$, and $\sigma_e^2 \sim IG(20, 0.2)$. Figure 1.3 shows the for one of the 100 data sets, the simulated real mean shape functions for the 50 subjects. These mean curves are various in phase and amplitude, which reflect the subject level variability.

To fit the functional model to the simulated data, we first conduct Bayesian curve registration to get estimation of the mean shape function for each subject. Figure 1.4 shows the mean shape function estimated by BHCR for the first 4 subjects. Compared with the true mean shape function, the estimated one has very close fit.

After achieving estimated mean function for each subject, we plug it into the functional regression model as the functional predictor, and run the regression model. To compare model performance we also fit the same model by penalized functional regression using the same set of predictors. Penalized functional regression is implemented by the R package "refund". For both

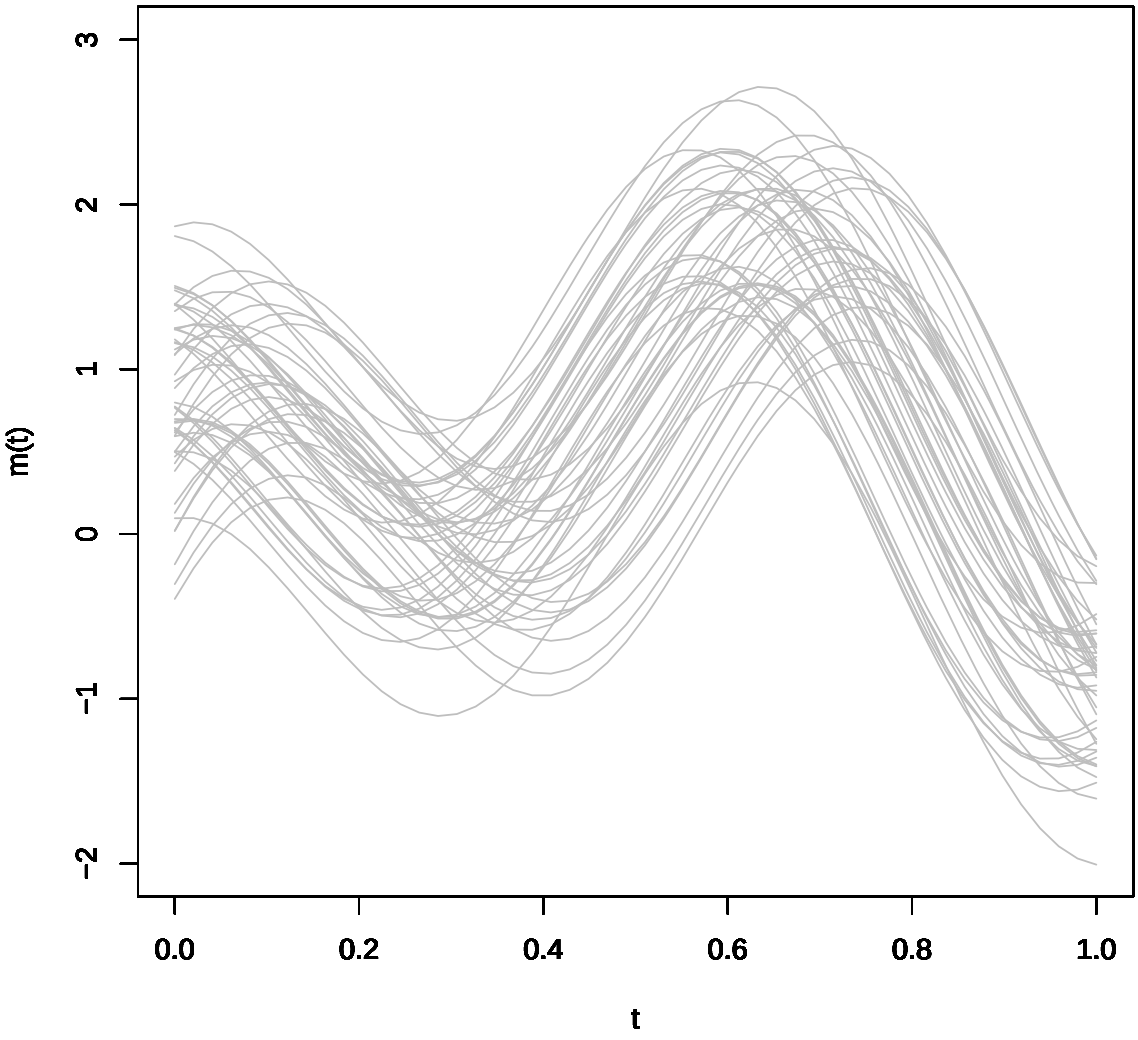

Figure 1.3: Simulated real mean shape functions

.

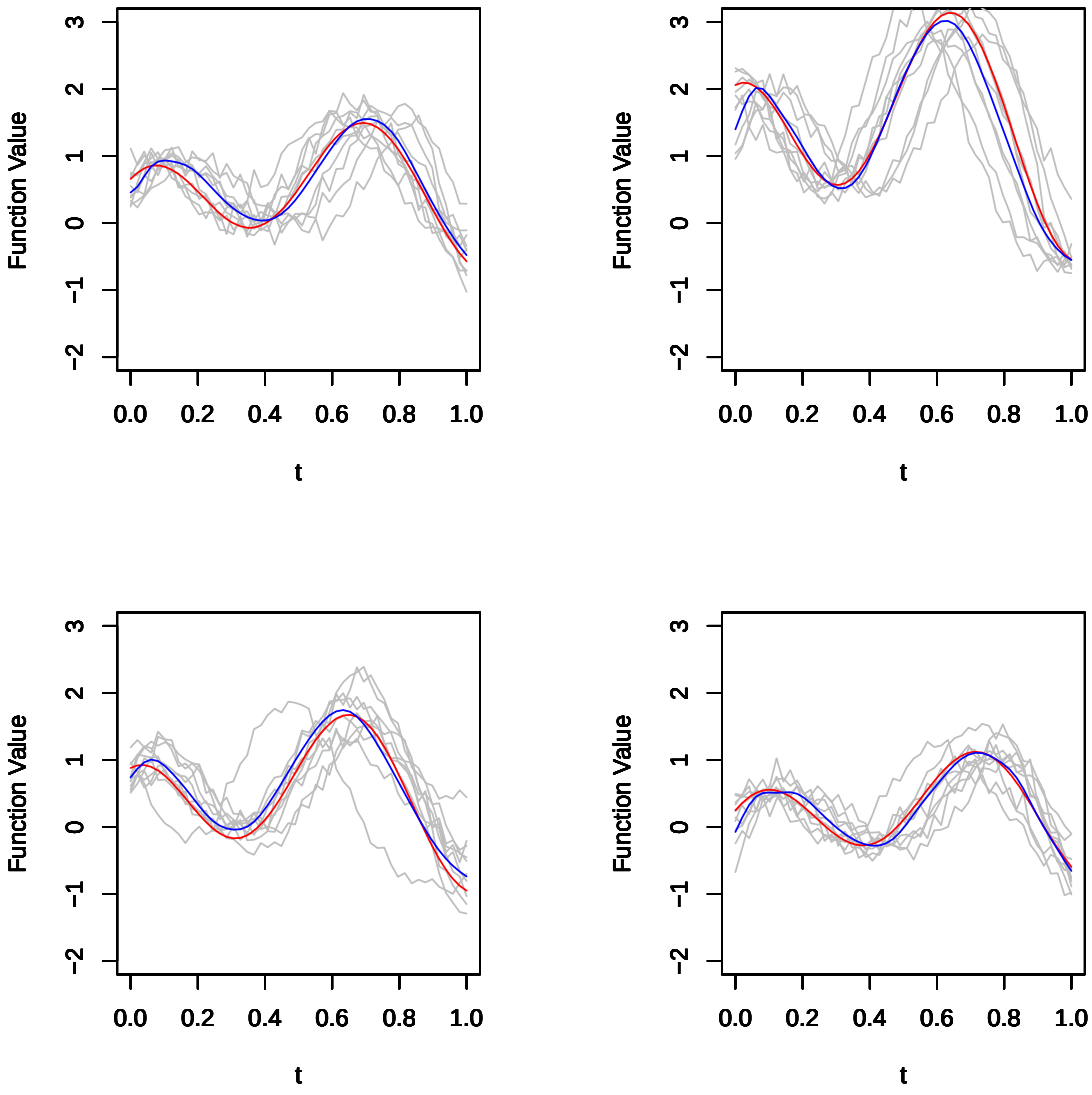

Figure 1.4: Simulated functional data for the first 4 subjects. Grey curves are the simulated functional data per each subject, blue curve is the estimated mean shape function, red curve is the true mean shape function.

Bayesian functional regression and penalized functional regression, we use the first order random walk as the penalization method for the functional coefficient $\boldsymbol{\alpha}(t)$, and we use 10 evenly distributed knots for its b-spline basis function. For Bayesian functional regression, we run MCMC 10000 times with the first 5000 runs as burn-in. Figure 1.5 shows the estimation results for one data set when the true functional coefficient is a continuous function denoted as $\boldsymbol{\alpha}_1(t)$. The two methods gave similar estimation results. The MSE for Bayesian functional regression and penalized functional regression to the true $\boldsymbol{\alpha}(t)$ is 0.012 and 0.005 respectively. Figure 1.6 shows the 95% confidence intervals of the estimated functional coefficient from the two methods based on the 100 data sets. The penalized functional regression has narrower CI than Bayesian functional regression in the middle area of the functional coefficient curve, but at the two ends the CI of penalized functional regression shows instability which implies the estimation is not reliable.

Figure 1.7 shows the estimation results when the true functional coefficient is a discontinuous function denoted as $\boldsymbol{\alpha}_2(t)$. For $\beta_2(t)$, Bayesian functional regression provided better estimation. The MSE for Bayesian functional regression and penalized functional regression to the true $\beta(t)$ is 0.024 and 0.036 respectively. Both of them failed to capture the feature that the functional coefficient is exactly zero in the area between 0.5 and 1. This is expected since they are using polynomial basis functions to approximate a linear relationship. However, the estimated functional coefficient by Bayesian model shows a much weaker relationship in the $(0.5, 1)$ area, which better estimates the trend in the true functional coefficient. On the contrary the penalized functional regression model over-smoothed the relationship in the first half of time range and underestimated the relationship in the second half of the time range. Figure 1.8 shows the 95% confidence intervals from the two methods. Same as the simulation above, the penalized functional regression has narrower CI than Bayesian functional regression in the middle area of the functional coefficient curve, but at the two ends the CI of penalized functional regression shows instability which implies the estimation is not reliable in those areas.

Binary Scalar Response Functional data and functional coefficient are simulated in the same way as in 1.3.2.3. We construct 100 repeated data sets. For each data set, we simulate 1000 subjects and 10 curves per subject. We also generate two types of functional coefficient: continuous and

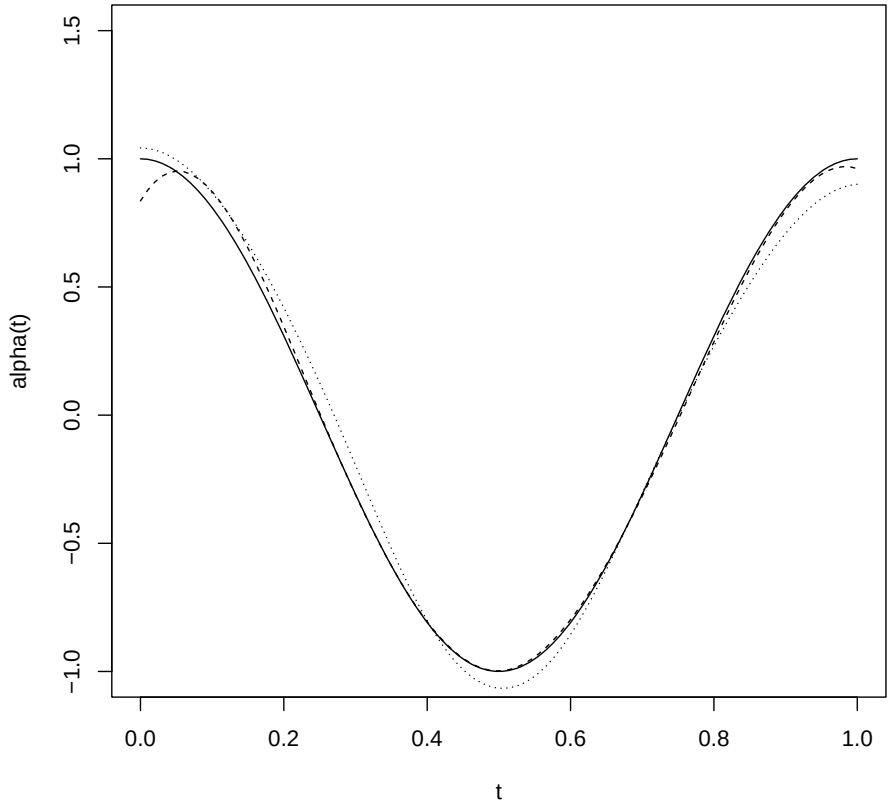

Figure 1.5: Estimation result for $\boldsymbol{\alpha}_1(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_1(t)$; dashed curve: estimated $\beta_1(t)$ by Bayesian functional regression; dotted curve: estimated $\boldsymbol{\alpha}_1(t)$ by penalized functional regression

.

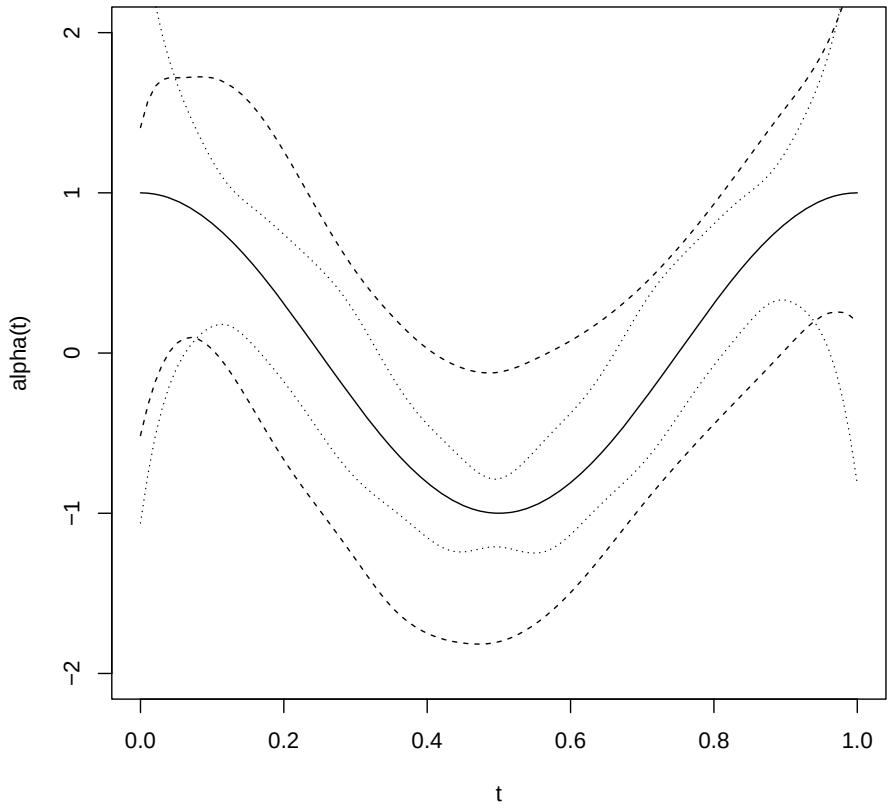

Figure 1.6: 95% Confidence interval for $\boldsymbol{\alpha}_1(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_1(t)$; dashed curve: CI by Bayesian functional regression; dotted curve: CI by penalized functional regression

.

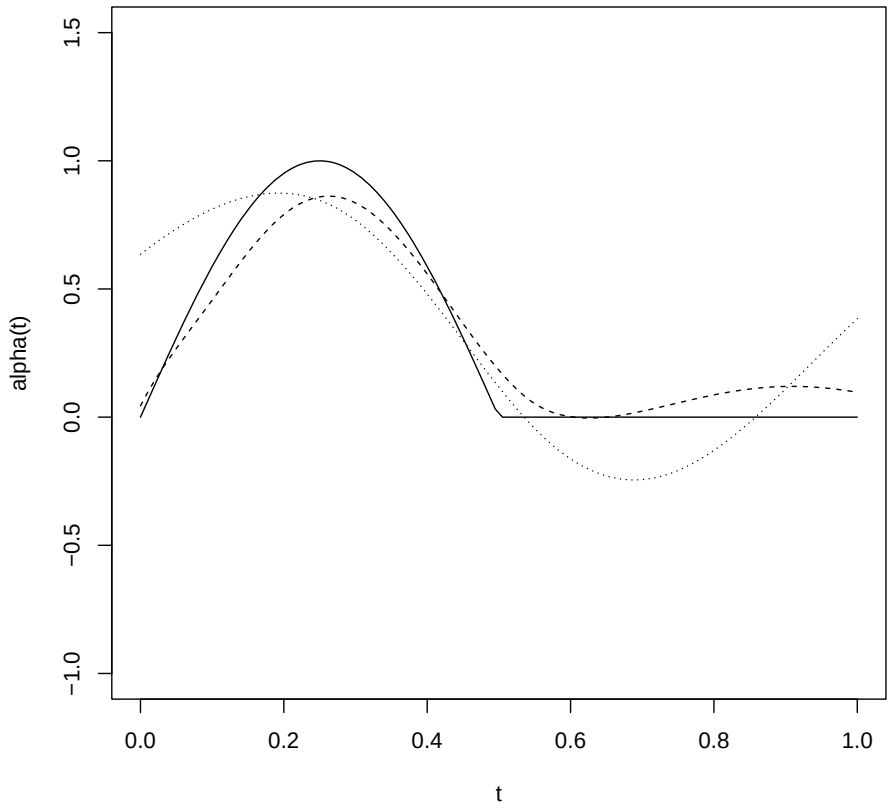

Figure 1.7: Estimation result for $\boldsymbol{\alpha}_2(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_2(t)$; dashed curve: estimated $\boldsymbol{\alpha}_2(t)$ by Bayesian functional regression; dotted curve: estimated $\boldsymbol{\alpha}_2(t)$ by penalized functional regression

.

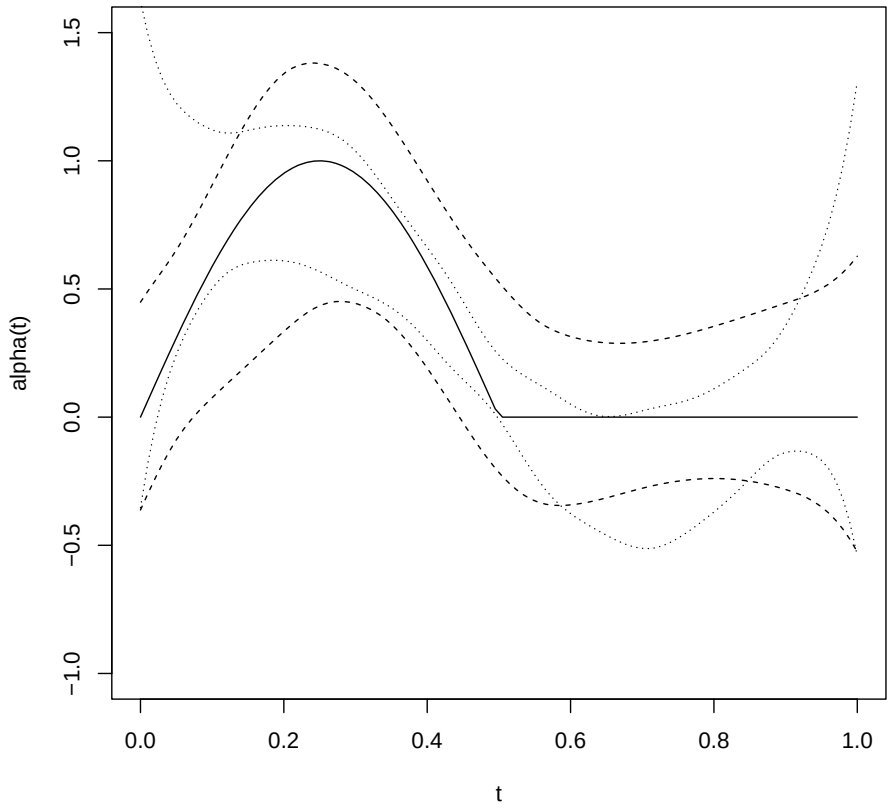

Figure 1.8: 95% Confidence interval for $\boldsymbol{\alpha}_2(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_2(t)$; dashed curve: CI by Bayesian functional regression; dotted curve: CI by penalized functional regression

.

discontinuous. Let

$$w_i = \sum_{s=1}^{100} m_i(t_s)\alpha(t_s)/100 + e_i \tag{1.73}$$

The binary outcome is generated by simulating p_i from $p_i = \phi(w_i)$ and $Y_i \sim$ Bernoulli(p_i). Figure 1.9 shows the estimation result for continuous functional coefficient scenario for one data set. Bayesian functional regression gives less satisfactory fitting at the beginning of the functional coefficient than for the rest part of the curve. Penalized functional regression method gives even poorer fitting result. It completely fails to capture the relationship between the binary outcome and functional predictor. For discontinuous functional coefficient scenario, both Bayesian and penalized functional regression are unable to provide appropriate estimation result based on Figure 1.10.

Application on DTI data Diffusion Tensor Imaging (DTI) is a magnetic resonance imaging based modality that traces the diffusion of water in the brain. Because water diffuses anisotropically in the white matter, DTI is able to generate images of the white matter specifically. Several measurements of water diffusion are provided by DTI, including fractional anisotropy and mean diffusivity. Then the summary of white matter tracts called tract profile can be derived from DTI. For neurodegeneration patients, such tract profile indicates the disease progress.

In a DTI study of multiple-sclerosis (MS), researchers hope to understand the relationship between cognitive disability and disease progress. The data set consists of 100 subjects, 66 women and 34 men, aged between 21 and 70 years at first visit. The number of visits per subject ranged from 2 to 8, with a median of 3, and were approximately annual; a total of 340 visits were recorded. At each visit full DTI scans were obtained and used to create tract profiles, accompanied by several tests providing scalar outcome of cognitive disability.

To apply Bayesian functional regression model on DTI data, we first estimate the mean shape function for each subject by curve registration, then use this mean function as the functional predictor, averaged cognitive score as the scalar outcome to establish a functional regression model. Figure 1.11 shows four patients with their original DTI tracts (in gray color) and estimated mean function. Figure 1.12 shows the 95% credible band of the estimated functional coefficient.

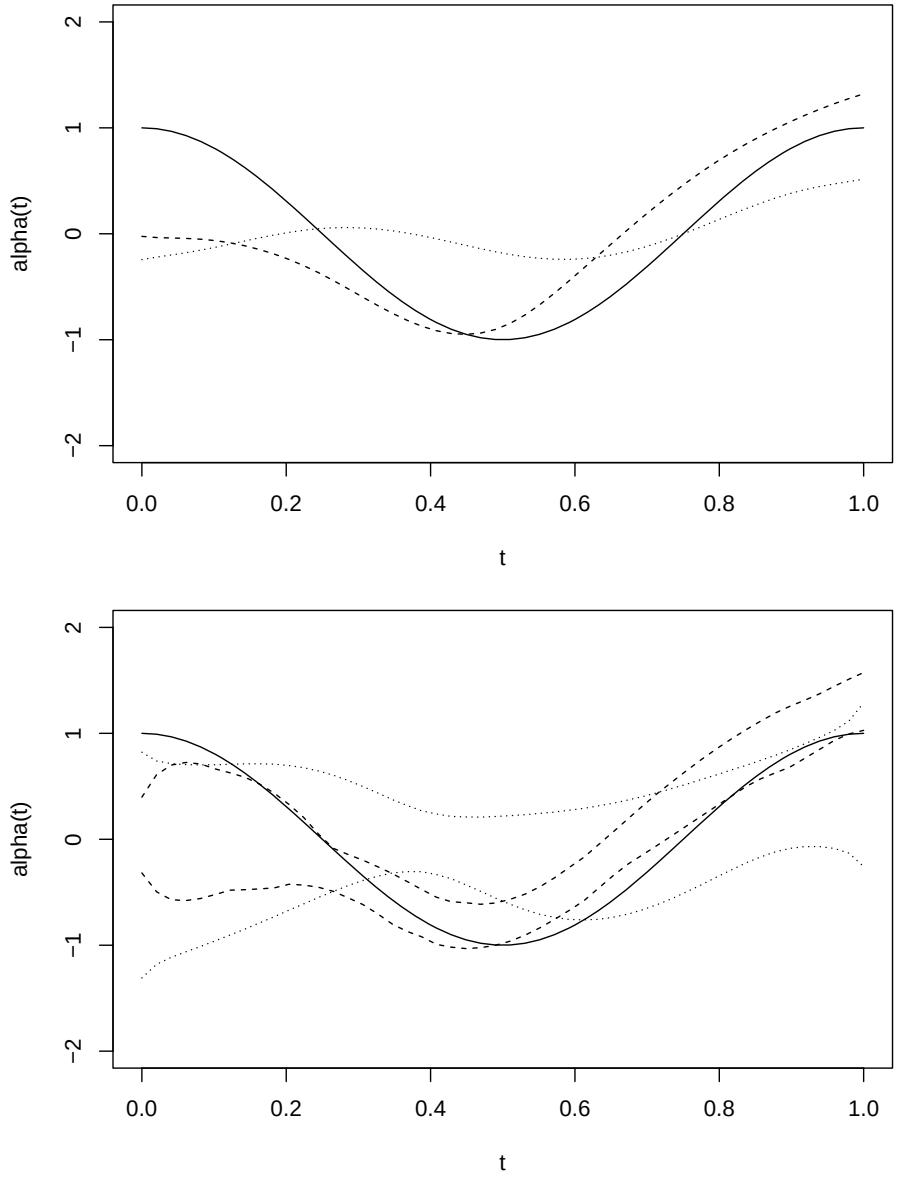

Figure 1.9: Continuous functional coefficient. Upper plot: Estimation result for $\boldsymbol{\alpha}_1(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_1(t)$; dashed curve: estimated $\boldsymbol{\alpha}_1(t)$ by Bayesian functional regression; dotted curve: estimated $\boldsymbol{\alpha}_1(t)$ by penalized functional regression. Lower plot: 95% Confidence interval for $\boldsymbol{\alpha}_1(t)$ by penalized functional regression and bayesian functional regression.

.

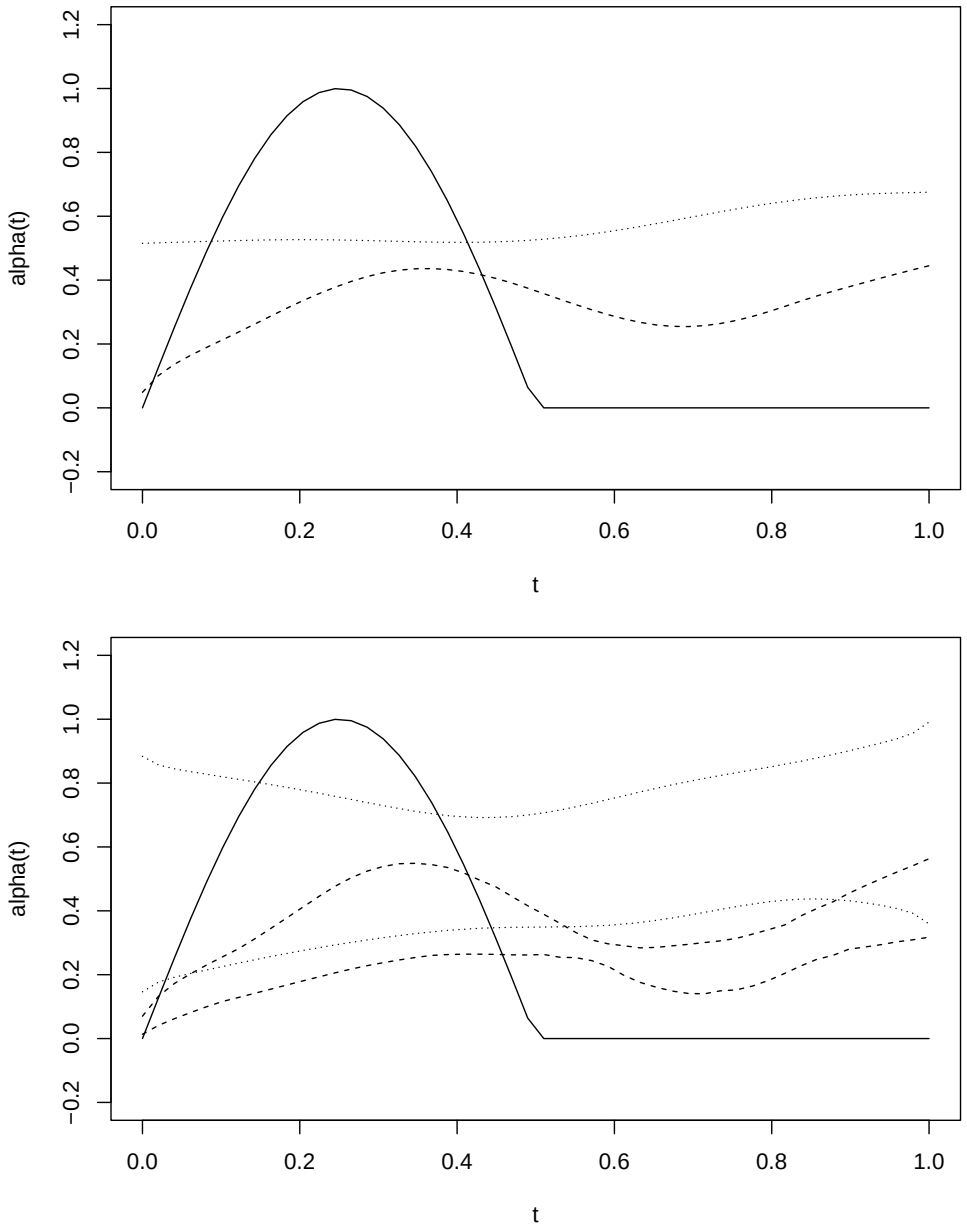

Figure 1.10: Discontinuous functional coefficient. Upper plot: Estimation result for $\boldsymbol{\alpha}_2(t)$ by penalized functional regression and bayesian functional regression. Solid curve: true $\boldsymbol{\alpha}_2(t)$; dashed curve: estimated $\boldsymbol{\alpha}_2(t)$ by Bayesian functional regression; dotted curve: estimated $\boldsymbol{\alpha}_2(t)$ by penalized functional regression. Lower plot: 95% Confidence interval for $\boldsymbol{\alpha}_2(t)$ by penalized functional regression and bayesian functional regression.

.

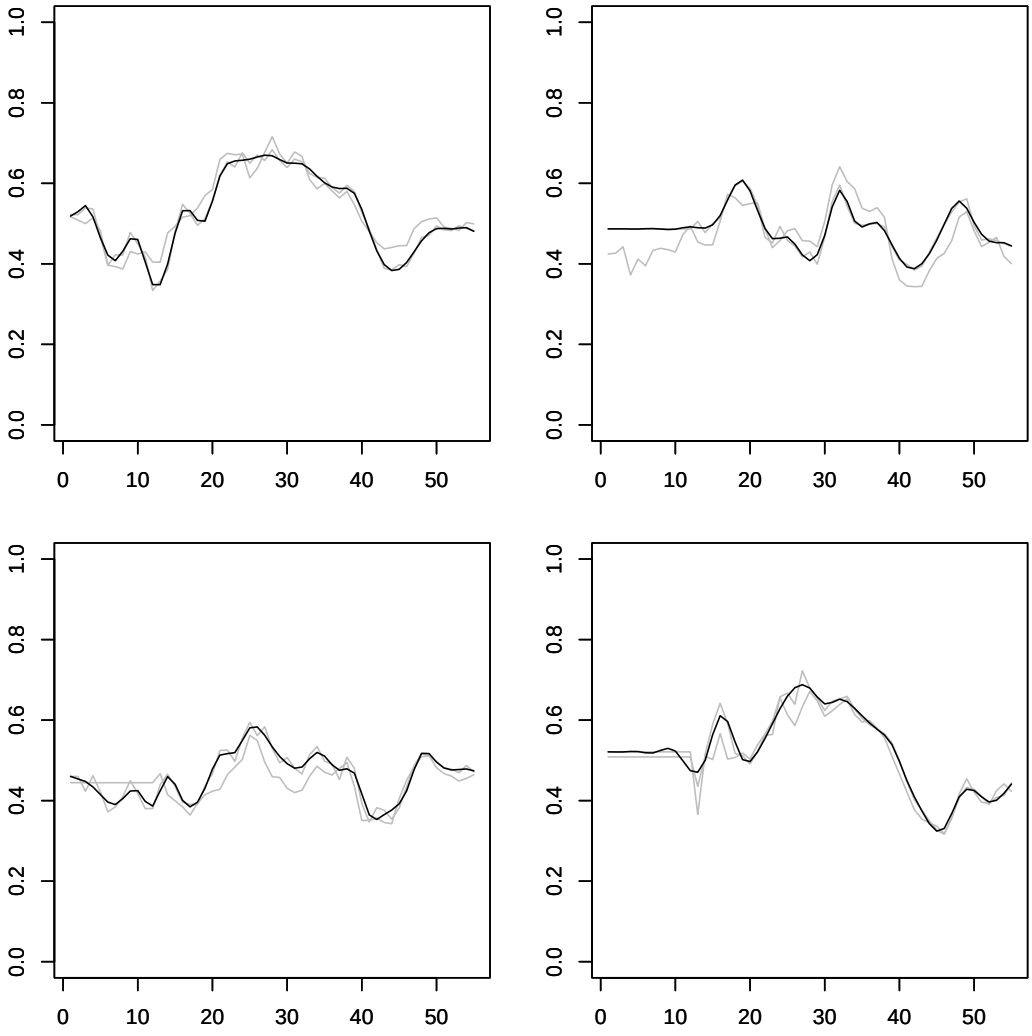

Figure 1.11: Profile mean function of four patients

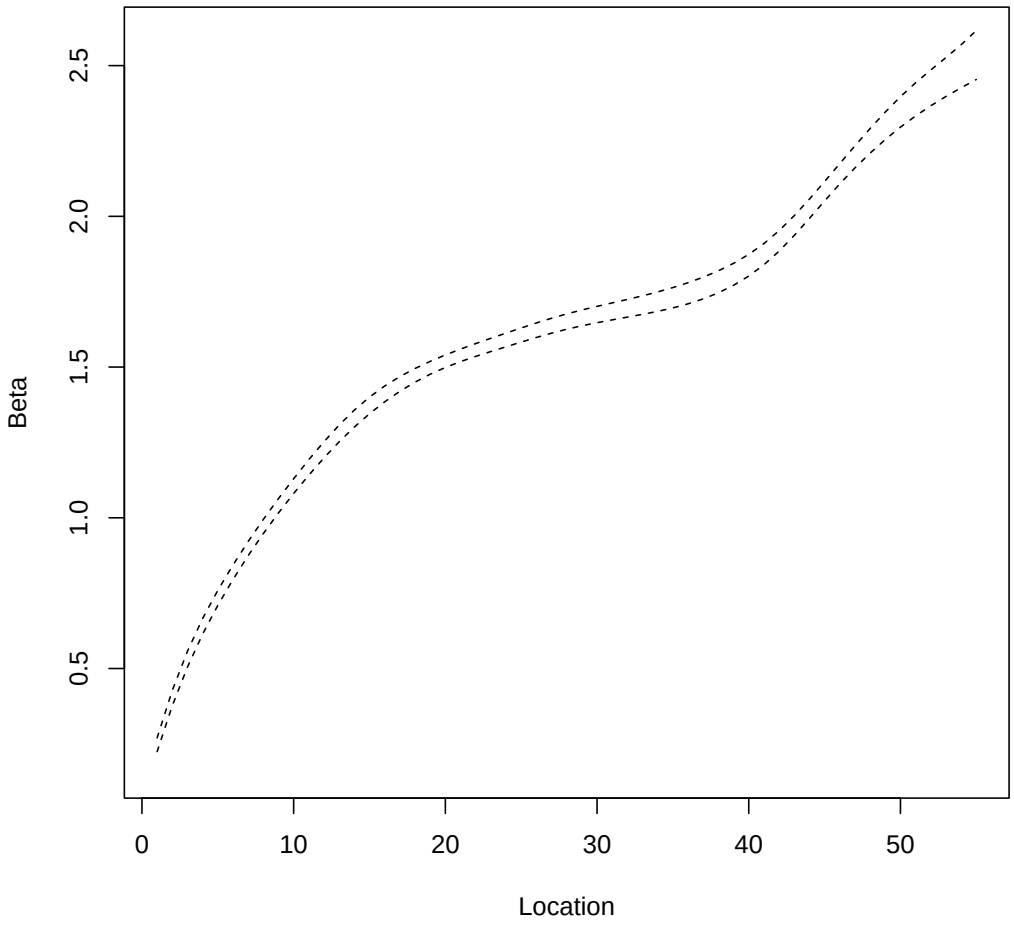

Figure 1.12: 95% credible band of estimated functional coefficient

Application on ICP Data As introduced in Chapter 4, ICP is an important diagnostic standard for neuro-surgical patients. In this application, we use functional regression model to investigate the relationship between patient's living status and their ICP curve. The data set contains 16 patients. Each patient has 100 to 200 ICP curves. We first use Bayesian curve registration process to obtain the mean ICP curve for each patient. Then we use the mean shape function as functional predictor in the regression model. Additionally, we also introduce two scalar predictors: the variation of scale parameter and amplitude parameter. Because we want to test if the variability among ICP curves of the same patient also plays a role in effecting the patient's living status. These two effects come from the curve registration step. The model is written as below:

$$\begin{aligned} \mathrm{E}(\boldsymbol{Y}) &= g(\boldsymbol{\eta}) \\ \boldsymbol{\eta} &= \boldsymbol{\gamma}_0 + \boldsymbol{H}_a\gamma_1 + \boldsymbol{H}_c\gamma_2 + \int \boldsymbol{M}(t)\boldsymbol{\nu}(t)dt + \boldsymbol{\epsilon} \end{aligned} \tag{1.74}$$

where $\boldsymbol{\gamma}_0$ is the intercept, $\boldsymbol{H}_a$ is the effect of amplitude parameter variability, and $\boldsymbol{H}_c$ is the effect of scale parameter variability. The mean ICP curves of different patients vary from $114/240$ minute to $238/240$ minute. We use 1 minute as the full time scale. For shorter than 1 minute curve, the missing part is made up with zero. The estimated functional coefficient is shown in Figure 1.13. Notice in the second half of the coefficient curve, the confidence band becomes very wide due to the fact many patients don't have data in this area, i.e., ICP curve is set to zero.

The 95% confidence interval of scale parameter variability is $[0.0029, 0.026]$, for amplitude parameter variability it is $[0.029, 0.069]$. Both of them do not cover zero which indicates the outcome have a significant relationship with the variability effect.

1.3.3 Functional Response Regression

Functional response regression is the regression of functional responses on a set of scalar predictors:

$$Y_i(t_j) = \sum_k X_{ik}B_k(t_j) + \epsilon_i(t_j) \tag{1.75}$$

where the functional coefficient $B_k(t_j)$ represents the effect of predictor X_k on response $Y_i(t_j)$ at time t_j. $\epsilon_i(t_j)$ is curve-to-curve residual error, whose covariance structure $C(t_1, t_2)$ describes

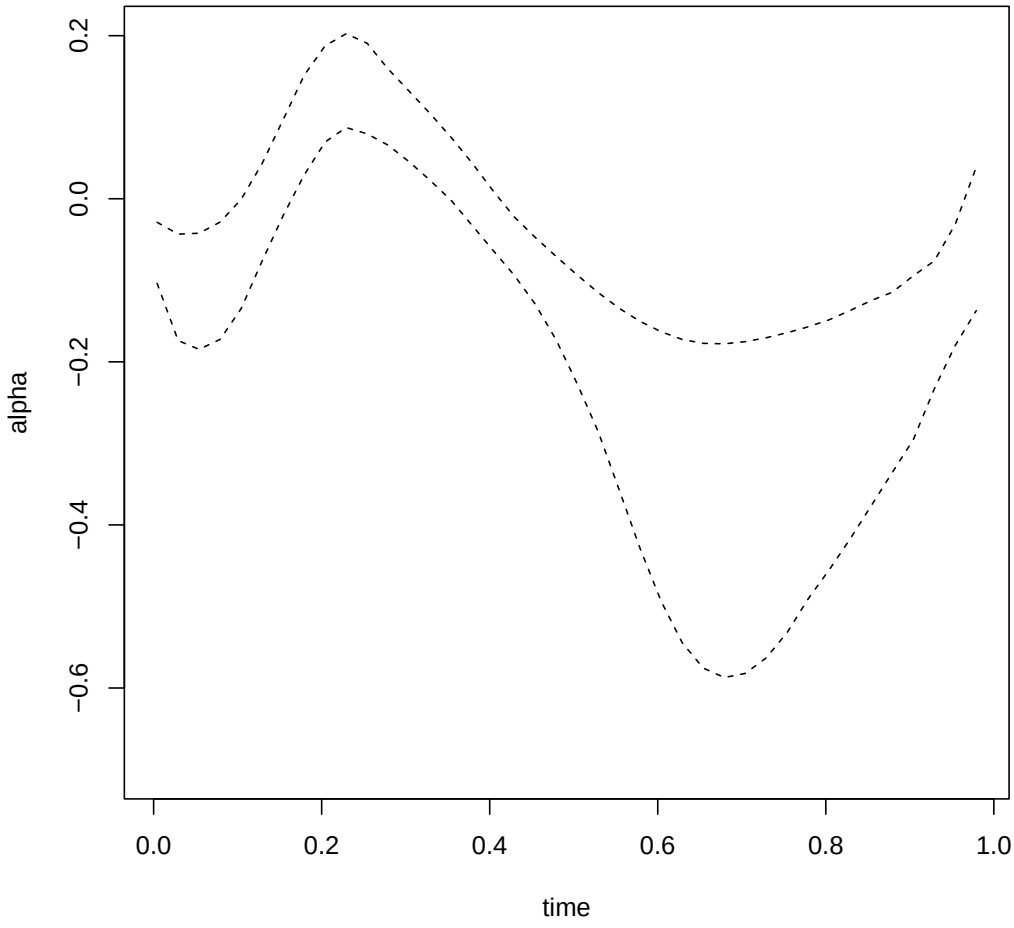

Figure 1.13: 95% credible band of estimated functional coefficient

the within function covariance. The goal of functional response regression is to estimate $B_k(t)$ and then test whether $B_k(t) = 0$. Among various frameworks proposed for functional response regression, functional mixed effects model is the one with most applications. [Morris and Carroll, 2006] introduced functional mixed effects model as follow:

$$Y_i(t_j) = \sum_k X_{ik} B_k(t_j) + \sum_h \sum_l Z_{ihl} U_{hl}(t_j) + \epsilon_i(t_j) \tag{1.76}$$

where h is the number of levels of random effects, Z_{ihl} are random effect covariates at level h with corresponding random effect functions $U_{hl}(t)$. $U_{hl}(t)$ are iid mean zero Gaussians with covariance $Q_h(t_1; t_2)$ representing the within-function covariance structure at random effect level h. These random effect functions can induce correlation between the functions through the structure of their design matrices. The model is fit by wavelet basis functions. [Bigelow and Dunson, 2007] introduced a Bayesian approach fitting the overall mean and curve-level random effect functions using truncated linear splines. [Thompson and Rosen, 2008] used B-spline to represent mean and random functions under Bayesian framework. [Fox and Dunson, 2012] presented a Bayesian approach that parameterizes the mean and random function using multi-resolution Gaussian processes.

This functional mixed effects model can be very flexible, therefore accommodating different scientific questions. The fixed effects can be mean functions, functional main effects, functional interactions, functional linear coefficients for continuous covariates, interactions of functional coefficients with other effects or any combination of these. The design matrix Z and between-curve correlation can be chosen to accommodate different covariance structures between curves that may be suggested by the experimental design. The random-effect portions of the model allow multiple hierarchical levels of random effects or to allow different random-effects distributions for different strata.

1.3.4 Function-on-Function Regression

Function-on-function regression can be formulated in the form:

$$Y_i(t) = B_0(t) + \int X_i(t) B(s,t) ds + \epsilon_i(t) \tag{1.77}$$

[Ramsay and Silverman, 2005] fit such model using basis function $\phi(s)$ and $\psi(t)$ for $X_i(s)$ and $Y_i(t)$ respectively. The estimator can be written as $B(s,t) = \phi(s)'\mathbf{B}\psi(t)$, where $\mathbf{B}$ is a K_x by K_y matrix containing the coefficient surface in the basis space.

[Yao et al., 2005] proposed functional PCA-based method for this model, where $Y(t)$ and $X(s)$ are modeled using functional PCA decomposition:

$$X(s) = \mu_x(s) + \sum_j \xi_j \phi_j(s) \tag{1.78}$$

$$Y(t) = \mu_y(t) + \sum_k \zeta_k \psi_k(t) \tag{1.79}$$

where $\mu_x(s) = E(X(s))$, $\mu_y(t) = E(Y(t))$. Their covariance functions are:

$$\begin{aligned} G_x(s_1, s_2) &= cov(X(s_1), X(s_2)) \\ G_y(t_1, t_2) &= cov(Y(t_1), Y(t_2)) \end{aligned}$$

Then the regression model becomes:

$$E(Y(t)|X) = \mu_y(t) + \int \beta(s,t)(X(s) - \mu_x(s))ds \tag{1.80}$$

Such model will be used in Chapter 3 and described in more details.

For function-on-function regression model, the relationship between response and predictor is controlled by the integration part. When it only integrates $s < t$ in the range of $X(s)$, such model is called historical regression model, since only historical values of $X(s)$ has impact on $Y(t)$. An extreme case is called concurrent regression model, where there is no integration in the model. That implies only current value of $X(t)$ has relationship with $Y(t)$, and $B(s,t)$ reduces to $B(t)$. This is a special case of varying coefficient model ([Hastie and Tibshirani, 1993]). Such model will also be discussed in more details in Chapter 3.

CHAPTER 2

Bayesian Curve Registration via Predictive Process Model

2.1 Introduction

In this chapter, we consider the problem of curve registration when each curve is observed over an intensive sampling grid. These data sources are becoming common in biomedical applications characterized by fine time resolution. Our motivating case is intracranial pressure (ICP) data analysis. ICP is monitored for patients in intensive care unit (ICU) who are suffering severe brain damage. It is a critical measurement for diagnosing and managing neuro-surgical patients. ICP data are in the form of pulses, where rich information related to cerebral pathophysiology embedded in the morphological feature. Raw ICP data are measured with artificial noises caused by active clinical environment. [Hu et al., 2009] established the Morphological Clustering and Analysis of ICP Pulse algorithm (MOCAIP). This algorithm is able to eliminate environment noises and extract a representative ICP pulse from an assigned time interval. Each pulse is sampled over intensive time grid. Depending on the length of the pulse, each pulse curve can have up to 200 sampling time points. During a 24-hour monitoring period, such representative ICP curves were extracted for every 5-min interval by MOCAIP. In our ICP data set, we have 16 patients and $100 \sim 200$ ICP pulses for each one (Figure 2.1). Within each patient, the variation in amplitude and phase between ICP pulses is observed. To obtain an overall ICP profile for every patient, it is necessary to do curve registration here.

Under general context of curve registration, we denote the observed values of the ith curve of a subject by vector: $\boldsymbol{y}_i = (y_{t_{i1}}, \ldots, y_{t_{ij}} \ldots, y_{t_{im_i}})'$, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$. Particularly, we are considering a curve registration problem when the number of time points m_i and number of curves n are both large.

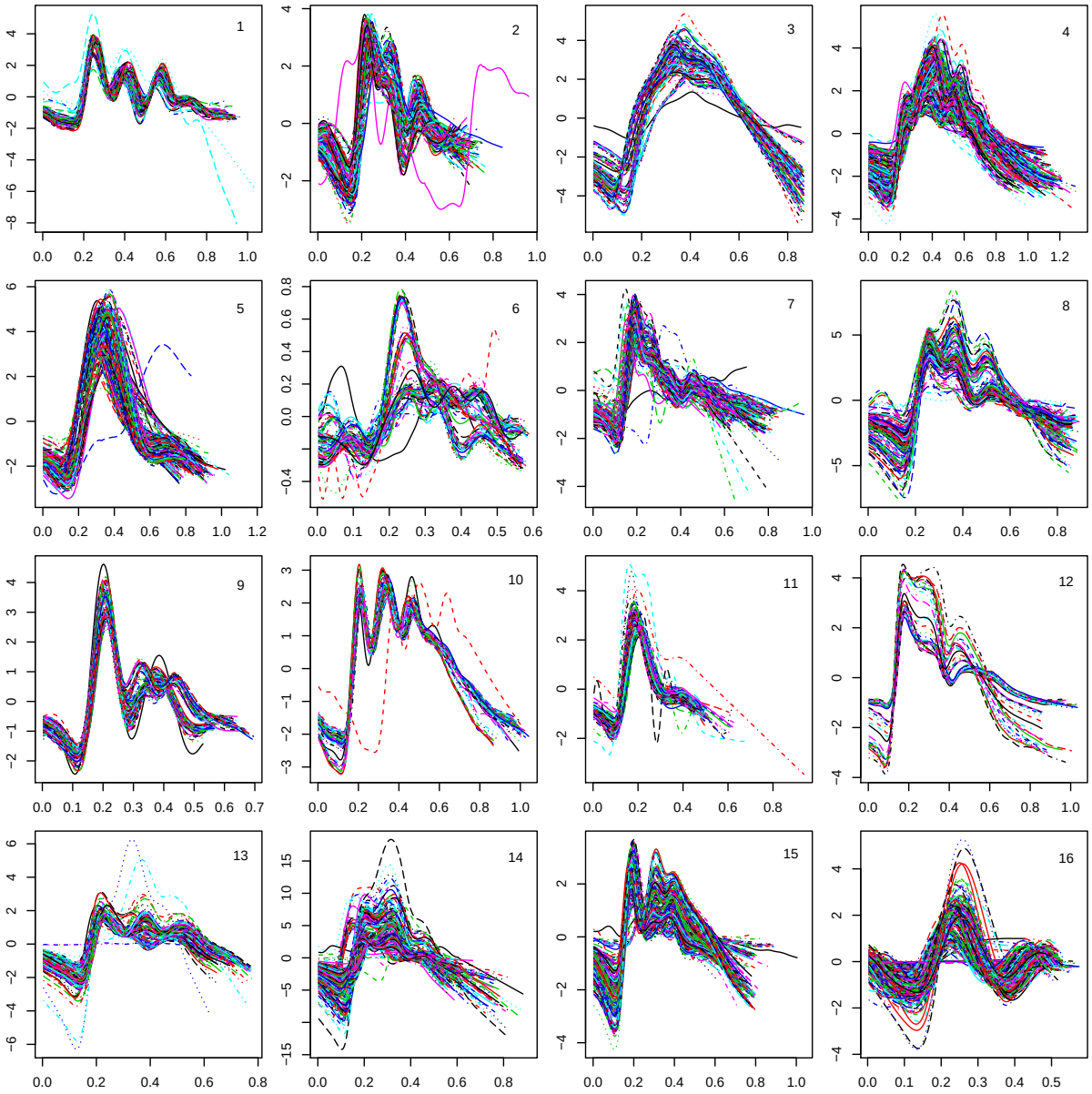

Figure 2.1: ICP Pulses Extracted by MOCAIP from 16 Patients with Brain Damage.

The standard curve registration setting outlined in [Telesca and Inoue, 2008] using MCMC technique is not computationally feasible in such case. In order to overcome computational limitation of MCMC algorithm, [Earls and Hooker, 2015] proposed an adapted variational Bayes procedure for approximate inference. This method is based on modeling functional data as Gaussian process. Since the parameter of time transformation function have non-conjugate priors (this is common for curve registration problem), the variational Bayes procedure is divided into two stages. The parameters with conjugate priors are estimated through standard variational Bayes formula, whereby the parameters with non-conjugate priors are estimated through maximizing a target function. The paper approved that this adapted variational Bayes method converges to an estimation where the parameters with conjugate priors are optimized as using standard variational Bayes method. However, a global optimization for all parameters is not guaranteed. While this technique is appealing from a computational perspective, it is likely to be most useful in exploratory analysis rather than formal inferential settings, as variational approximations are notoriously unsatisfactory in the characterization of uncertainty. To conquer the computational challenge and provide formal statistical inference, we propose predictive process model (PPM) for curve registration, inspired by the literature in Spatial statistics [Banerjee et al., 2008]. To authors' knowledge this is the first time an acceleration algorithm for Bayesian curve registration is proposed without sacrificing formal inference property from MCMC simulation.

This chapter is organized as follows. Section 2 introduces the model formulation and prior settings of this novel registration method. Section 3 describes posterior simulation and inference via MCMC. Section 4 describes an acceleration method through fixing one of the parameter value. In Section 5, the proposed methods are applied to simulated data and compared with existing standard method. In Section 6 we apply the proposed method to ICP data, which are computationally challenging when using standard method. Finally in Section 7 we provide a discussion.

2.2 Model Formulation

2.2.1 Predictive Process Model

The standard Bayesian hierarchical curve registration (BHCR) considers the following representation of a functional process

$$y_i(t) = c_i + a_i\, m(t) \circ \mu_i(t) + \epsilon_i(t).$$

Basis function representation of the functional convex mean $m(t) = \mathcal{B}'_m(t)\boldsymbol{\beta}$ and time transformation function $\mu_i(t) = \mathcal{B}'_\mu(t)\boldsymbol{\phi}_i$ are sought, where $\mathcal{B}_m(t)$ and $\mathcal{B}_\mu(t)$ are vectors of basis functions evaluated at time t, $\boldsymbol{\beta}$ and $\boldsymbol{\phi}_i$ are the coefficient vectors. Then the model above is written as

$$y_i(t) = c_i + a_i\, \mathcal{B}'_m(t)\beta \circ \mathcal{B}'_\mu(t)\phi_i + \epsilon_i(t).$$

Under this setting, assuming the functional process is observed with white noise $\epsilon_i(t) \sim N(0, \sigma^2_\epsilon)$, posterior simulation is standard and, in principle, easy to implement by MCMC algorithm. The conditional posterior distributions of both $\boldsymbol{\beta}$ and $\boldsymbol{\phi}_i$, however, rely on the re-calculation of the basis expansion $\mathcal{B}_m(\mu_i(t))$ as one updates $\mu_i(t)$ in each transition of the Markov Chain, leading to the reconfiguration of the design matrix needed to define the conditional posterior mean of $\boldsymbol{\beta}$, with a minimal flop count of order $O(n^2(\sum_i m_i)^2)$ in each MCMC iteration. Furthermore, updates of $\boldsymbol{\phi}_i$, $(i = 1, \ldots, n)$ require a minimal number of operations of order $O(\sum_i m_i)$ in each iteration. When m_i is large, this procedure is likely to be unfeasible on most computational platforms.

This observation motivates an alternative parameterization of the curve registration problem, using predictive process models as follows. Define:

$$y_i(t) = f_i(t) + \epsilon_i(t), \text{ with } f_i(t) = \mathcal{B}'_f(t)\boldsymbol{b}_i. \tag{2.1}$$

If the expansion of $f_i(t)$ is based on k basis functions, then $\boldsymbol{b}_i = (b_{i1}, \ldots, b_{ik})'$.

The problem of curve registration can be re-formulated by representing $\boldsymbol{b}_i$ as the realization of a registration process $\gamma_i(\tau)$, defined as

$$\gamma_i(\tau) = c_i + a_i\, m(\tau) \circ \mu_i(\tau) + \delta_i(\tau) = c_i + a_i \mathcal{B}'_m(\tau)\boldsymbol{\beta} \circ \mathcal{B}'_\mu(\tau)\boldsymbol{\phi}_i + \delta_i(\tau).$$

Note that $\boldsymbol{b}_i$ is not directly indexed by time, however we may proceed using the following convention. We consider an arbitrary domain for the process $\gamma_i(\tau)$, so that $\tau \in [0,1]$. Furthermore, we let $\boldsymbol{b}_i \sim N(\tilde{\boldsymbol{\gamma}}_i, g\sigma_\epsilon^2\mathbf{I})$, where $\tilde{\boldsymbol{\gamma}}_i$ is the process $\boldsymbol{\gamma}_i$ conventionally evaluated at a grid of k points in $[0,1]$, so that $\tilde{\boldsymbol{\gamma}}_i = (\gamma_i(0), \gamma_i(\frac{1}{k-1}), \gamma_i(\frac{2}{k-1}), \ldots, \gamma_i(\frac{j-1}{k-1}), \ldots, \gamma_i(1))'$.

This model, while seemingly overparametrized when compared to the standard hierarchical curve registration framework, leads to significant computational savings when assuming $k << \min_i(m_i)$. Posterior simulation is still standard, but now only requires the following set of operations:

1) $n\, O(m_i^2)$ for the computation of the conditional posterior for $\boldsymbol{b}_i$, $i = 1, \ldots, n$. Note that this calculation only needs to be carried out once.

2) $(n\, k)^2$ for the computation of the conditional posterior for $\boldsymbol{\beta}$ at each iteration.

3) $O(nk)$ for the computation of the conditional posterior for $\boldsymbol{\phi}_i$ at each iteration.

To illustrate how this predictive process model based curve registration (PPM-BHCR) compared to the original Bayesian hierarchical curve registration (BHCR), we omit t from the expression and let $\boldsymbol{y}_i$ denote the vector of observations of the ith subject. For PPM-BHCR, multiplying matrix $(\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'$ on both sides of the equation, we obtain:

$$(\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'\boldsymbol{y}_i = c_i\mathbf{1} + a_i\mathcal{B}_m(\mathcal{B}_\mu(\tau)\boldsymbol{\phi}_i)\boldsymbol{\beta} + \delta_i\mathbf{1} + (\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'\boldsymbol{\epsilon}_i \tag{2.2}$$

Let $\boldsymbol{z}_i = (\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'\boldsymbol{y}_i$ denote the linear transformed vector of observations, and let $\boldsymbol{w}_i = \delta_i\mathbf{1} + (\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'\epsilon_i$ denote the vector of error terms of the new observation vector, then the PPM-BHCR process is realized by the following equation:

$$\boldsymbol{z}_i = c_i\mathbf{1} + a_i\mathcal{B}_m(\mathcal{B}_\mu(\tau)\boldsymbol{\phi}_i)\boldsymbol{\beta} + \boldsymbol{w}_i \tag{2.3}$$

$\boldsymbol{z}_i$ is the same functional process assessed on different time grid. If we normalize the original time grid of $\boldsymbol{y}_i$ to the same range as τ, then the time transformation functions from the two registration processes are the same. For the purpose of acceleration, we set $k << m$. Therefore $\boldsymbol{z}_i$ is a functional observation with much more sparse sampling grid than $\boldsymbol{y}_i$.

Figure 2.2 is an example showing how the latent process $\gamma(\tau)$ generates phase varying curves. $\gamma_i(\tau)$ is simulated by $c_i + a_i\, m(\tau) \circ \mu_i(\tau) + \delta_i(\tau)$, where the time transformation function $\mu_i(\tau)$

is Beta cumulative density function, and shape function is $m(t) = sin(15t)exp(-10(t-0.5)^2)$. y is generated by $y_i(t) = \mathcal{B}_f'(t)\boldsymbol{b}_i$, where $\mathcal{B}_f'(t)$ is the design matrix of basis functions, and $\boldsymbol{b}_i$ is realization of the registration process $\gamma_i(\tau)$. For simplicity purpose, we omit the error terms in the example below. By using B-spline basis function, the registration process works well to generate functional outcome with time variation. However, when using PCA basis function, the registration process fails to generate outcome whose time variation can be recovered from $\gamma_i(\tau)$.

2.2.2 Prior and Basis Function Setting

This predictive process model based Bayesian curve registration is hierarchical and has three levels:

Level One. The observed function data are modeled as: $y_i(t) = f_i(t) + \epsilon_i(t) = \mathcal{B}_f(t)'\boldsymbol{b}_i + \epsilon_i(t)$ where $\epsilon_i(t) \sim N(0, \sigma_\epsilon^2)$. Assign Gaussian prior on $\boldsymbol{b}_i$:

$$\boldsymbol{b}_i \sim N(\tilde{\boldsymbol{\gamma}}_i, g\sigma_\epsilon^2\mathbf{I}), \tag{2.4}$$

where g is a tuning parameter. Such prior is similar with the g-prior by [Zellner, 1986]. g determines how much variation in $\boldsymbol{b}_i$ comes from random errors in y. When g goes large, the posterior mean of $\boldsymbol{b}_i$ will be close to its maximum likelihood estimator. Ideally, the estimation of registration process should be robust to different choices of g value, therefore g value can be prespecified instead of being treated as a hyper-parameter. A sensitivity analysis of g is conducted in simulation study 4 and it shows the robustness of g in the proposed method.

Level Two. On this level, $\gamma_i(\tau)$ is modeled as $\gamma_i(\tau) = c_i + a_i\, m(\tau) \circ \mu_i(\tau) + \delta_i(\tau) = c_i + a_i\mathcal{B}_m'(\tau)\boldsymbol{\beta} \circ \mathcal{B}_\mu'(\tau)\boldsymbol{\phi}_i + \delta_i(\tau)$. Gaussian priors are assigned to these two parameters:

$$\begin{aligned} c_i &\sim N(0, \sigma_c^2) \\ a_i &\sim N(1, \sigma_a^2)I(a_i > 0) \end{aligned} \tag{2.5}$$

For $\mu_i(\tau)$, it needs to be strictly monotonically increasing and confined on the support of τ: $\tau_1 \leq \mu_1(\tau) < \mu_2(\tau) < ... < \mu_n(\tau) \leq \tau_k$. Multivariate Gaussian prior is used on the coefficient $\boldsymbol{\phi}_i$:

$$\boldsymbol{\phi}_i \sim N(\boldsymbol{\Upsilon}, \lambda_\phi\boldsymbol{\Omega}_\phi^{-1}) \tag{2.6}$$

where $\boldsymbol{\Upsilon}$ is coefficient for the identity transformation function which satisfies $\mu(\tau; \boldsymbol{\Upsilon}) = \tau$. For

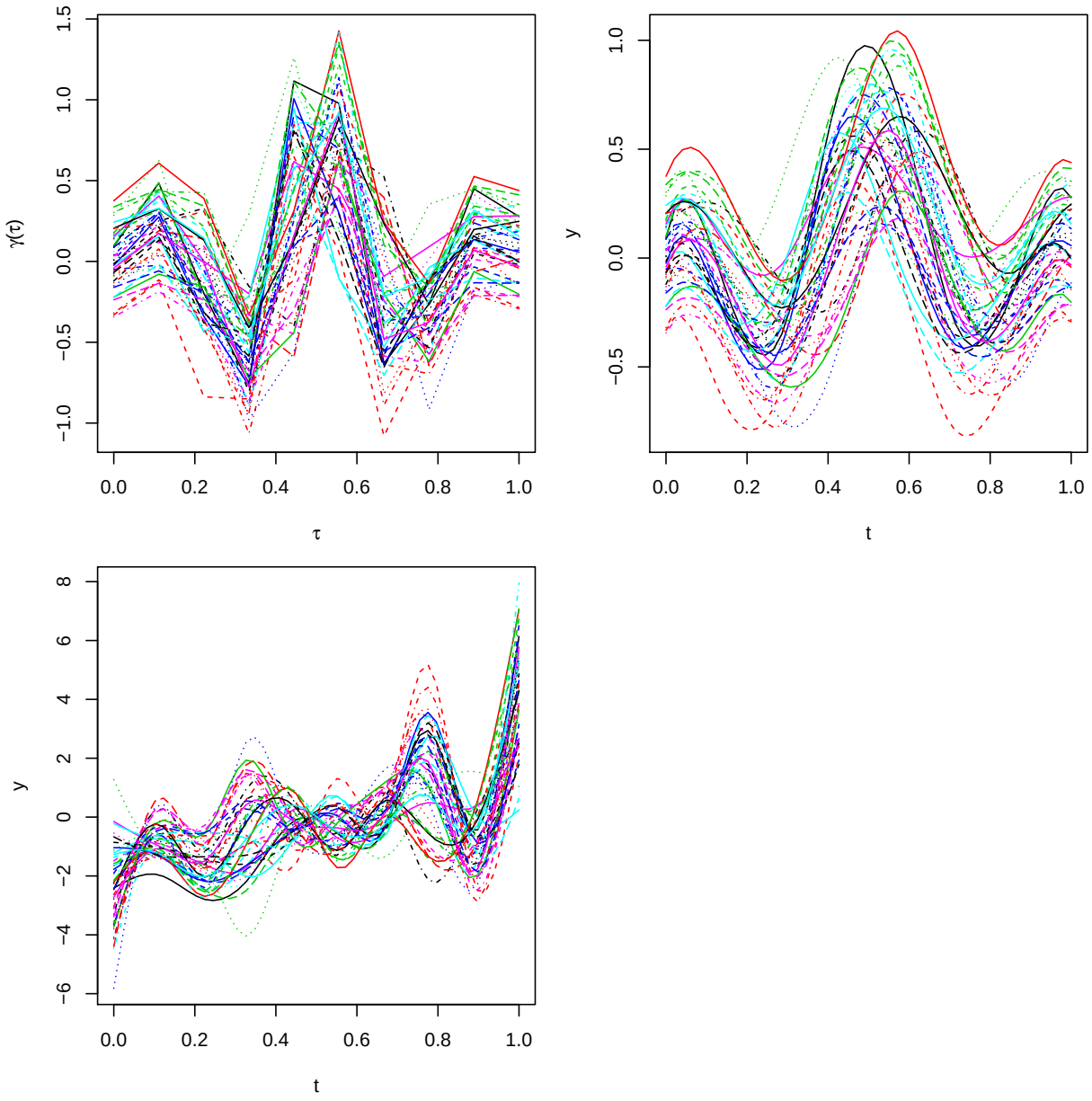

Figure 2.2: Registration process example. Upper left: $\gamma(\tau)$. Upper right: $\gamma(\tau)$ generated y using B-spline basis. Lower left: $\gamma(\tau)$ generated y using PCA basis.

the coefficient $\boldsymbol{\beta}$, it is also assigned multivariate Gaussian prior:

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \lambda_\beta \boldsymbol{\Omega}_\beta^{-1}) \tag{2.7}$$

The estimation of $\boldsymbol{\phi}$ and $\boldsymbol{\beta}$ is a curve-fitting problem. Their smoothness is controlled through P-spline ([Lang and Brezger, 2004]), where the second order random walk priors are used. $\boldsymbol{\Omega}_\beta$ and $\boldsymbol{\Omega}_\phi$ are special banded matrices in the form below. λ_β and λ_ϕ are smoothing parameters.

$$\boldsymbol{\Omega} = \begin{pmatrix} 6 & -4 & 1 & 0 & & & & 0 \\ -4 & 6 & -4 & 1 & \ddots & & & \\ 1 & -4 & 6 & -4 & 1 & \ddots & & \\ 0 & 1 & \ddots & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & 1 & -4 & 6 & -4 & 1 \\ & & & \ddots & 1 & -4 & 5 & -2 \\ 0 & & & & \ddots & 1 & -2 & 1 \end{pmatrix}$$

Level Three. Priors on hyperparameters:

$$\begin{aligned} 1/\sigma_a^2 &\sim Gamma(a_a, b_a) \\ 1/\sigma_c^2 &\sim Gamma(a_c, b_c) \\ 1/\sigma_\epsilon^2 &\sim Gamma(a_\epsilon, b_\epsilon) \\ 1/\lambda_\phi &\sim Gamma(a_\lambda, b_\lambda) \\ 1/\lambda_\beta &\sim Gamma(a_\lambda, b_\lambda) \end{aligned} \tag{2.8}$$

2.3 Posterior Simulation and Inference

Let $\boldsymbol{\theta} = (\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{b}', \boldsymbol{\beta}', \boldsymbol{\phi}', \sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \lambda_\phi, \lambda_\beta)$ denote the vector of all parameters, then the posterior distribution is:

$$\begin{aligned} f(\boldsymbol{\theta}|\boldsymbol{Y}) &= f(\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{b}', \boldsymbol{\beta}', \boldsymbol{\phi}', \sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \lambda_\phi, \lambda_\beta) \\ &\propto f(\boldsymbol{Y}|\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{b}', \boldsymbol{\beta}', \boldsymbol{\phi}', \sigma_\epsilon^2) f(\boldsymbol{c}, \boldsymbol{a}|\sigma_c^2, \sigma_a^2) \\ &\quad \times f(\boldsymbol{b}|\sigma_\epsilon^2) f(\boldsymbol{\beta}|\lambda_\beta) f(\boldsymbol{\phi}|\lambda_\phi) \\ &\quad \times f(\sigma_\epsilon^2|a_\epsilon, b_\epsilon) f(\sigma_c^2|a_c, b_c) f(\sigma_a^2|a_a, b_a) \\ &\quad \times f(\lambda_\beta|a_\lambda, b_\lambda) f(\lambda_\phi|a_\lambda, b_\lambda) \end{aligned} \tag{2.9}$$

Since the joint distribution is intractable to directly sample from, then MCMC sampling method is applied to draw samples from it. Except $\boldsymbol{\phi}$, all the other parameters have standard full conditional posterior distribution since they have conjugate priors. For $\boldsymbol{c}', \boldsymbol{a}', \boldsymbol{\beta}', \boldsymbol{b}'$, their full conditional posterior distribution is Gaussian distribution. For $\sigma_\epsilon^2, \sigma_c^2, \sigma_a^2, \lambda_\phi, \lambda_\beta$, their full conditional distribution is inverse Gamma. So Gibbs sampler is suitable to draw samples from these distributions. For $\boldsymbol{\phi}$, $\gamma(\tau)$ is nonlinear in $\boldsymbol{\phi}$ and therefore the prior is non-conjugate. Metropolis-Hasting algorithm is adopted to simulate samples from the posterior of $\boldsymbol{\phi}$. The proposal density is calibrated so that the acceptance rate for Metropolis-Hasting algorithm is between 0.25 to 0.75. In summary, this MCMC sampling method is a mixture of Gibbs sampler and Metropolis-Hasting algorithm. The detailed posterior functions are listed in appendix. The algorithm is as below:

1. Simulate samples from posterior distributions for parameters with conjugate prior.
2. For $i = 1, ..., N$, simulate sample for $\boldsymbol{\phi}_i$ from its posterior by;
   (a) For $j = 1, ..., Q$:
      i. propose ϕ_{ij}^* from its support.
      ii. Calculate the posterior ratio $r = \frac{p(b_i|\boldsymbol{\phi}_i^*, \boldsymbol{\theta}_{-\phi_i}) p(\boldsymbol{\phi}_i^*)}{p(b_i|\boldsymbol{\phi}_i, \boldsymbol{\theta}_{-\phi_i}) p(\boldsymbol{\phi}_i)}$
      iii. Accept $\boldsymbol{\phi}_{ij}^*$ with probability $min(1, r)$.
3. Iterate the last two steps.

Before sampling, time grid t needs to be normalized to the same domain as τ. After M draws of simulated samples, the time transformation function for the ith subject $\mu_i(t)$ is calculated by:

$$\bar{\mu_i}(t) \quad = \quad \mathcal{B}_\mu(t)\bar{\boldsymbol{\phi}}_i \tag{2.10}$$

where $\bar{\boldsymbol{\phi}}_i$ is the average over the M simulated samples. $\boldsymbol{\phi}_i$ is the coefficient for $\mu(\tau)$. $\mathcal{B}_\mu(t)$ is a compatible design matrix evaluated over t. t and τ have the same domain. The estimated mean function is calculated by the cross-sectional mean of aligned functions.

Because m can be very large, the posterior variance of $\boldsymbol{b}_i$ is close to zero. Therefore, to further accelerate the PPM-BHCR algorithm, we fix $\boldsymbol{b}_i$ at its least square solution:

$$\hat{\boldsymbol{b}}_i = (\mathcal{B}_f'\mathcal{B}_f)^{-1}\mathcal{B}_f'\boldsymbol{y}_i \tag{2.11}$$

In MCMC simulation we can skip the simulation of $\boldsymbol{b}_i$ and use $\hat{\boldsymbol{b}}_i$ when $\boldsymbol{b}_i$ is needed in simulation of other parameters.

2.4 Simulation Studies

2.4.1 Simulation Study 1: Evaluate Model Fit of PPM-BHCR

To assess the estimation by the predictive process model based curve registration, we simulated one data set as following. In this data set we simulated $N = 50$ random curves with each curve generated by $y_i(t) = c_i + a_i m(\mu(t)) + \epsilon_i,\ i = 1, ..., 50$. The common shape function is: $m(t) = sin(15t)exp(-10(t-0.5)^2)$. Time transformation function is $f(t; \alpha_1, \alpha_2)$ which is the cumulative density function of Beta distribution. α_1 and α_2 are random samples from $Gamma$(50,50). The time grid has $m = 500$ equally spaced time points.

To apply the predictive process model based curve registration, we set $k = m/10$, and use relatively diffuse $Gamma(.1, .1)$ prior for the precision parameter of $\boldsymbol{\beta}$ and $\boldsymbol{\phi}_i$. We run the MCMC simulation 2000 times with the first 1000 iterations as burn-in.

Simulation results are displayed in Figure 2.3. It shows compared to true alignment, PPM-BHCR offers satisfactory performance on registering functions with both amplitude and phase variability.

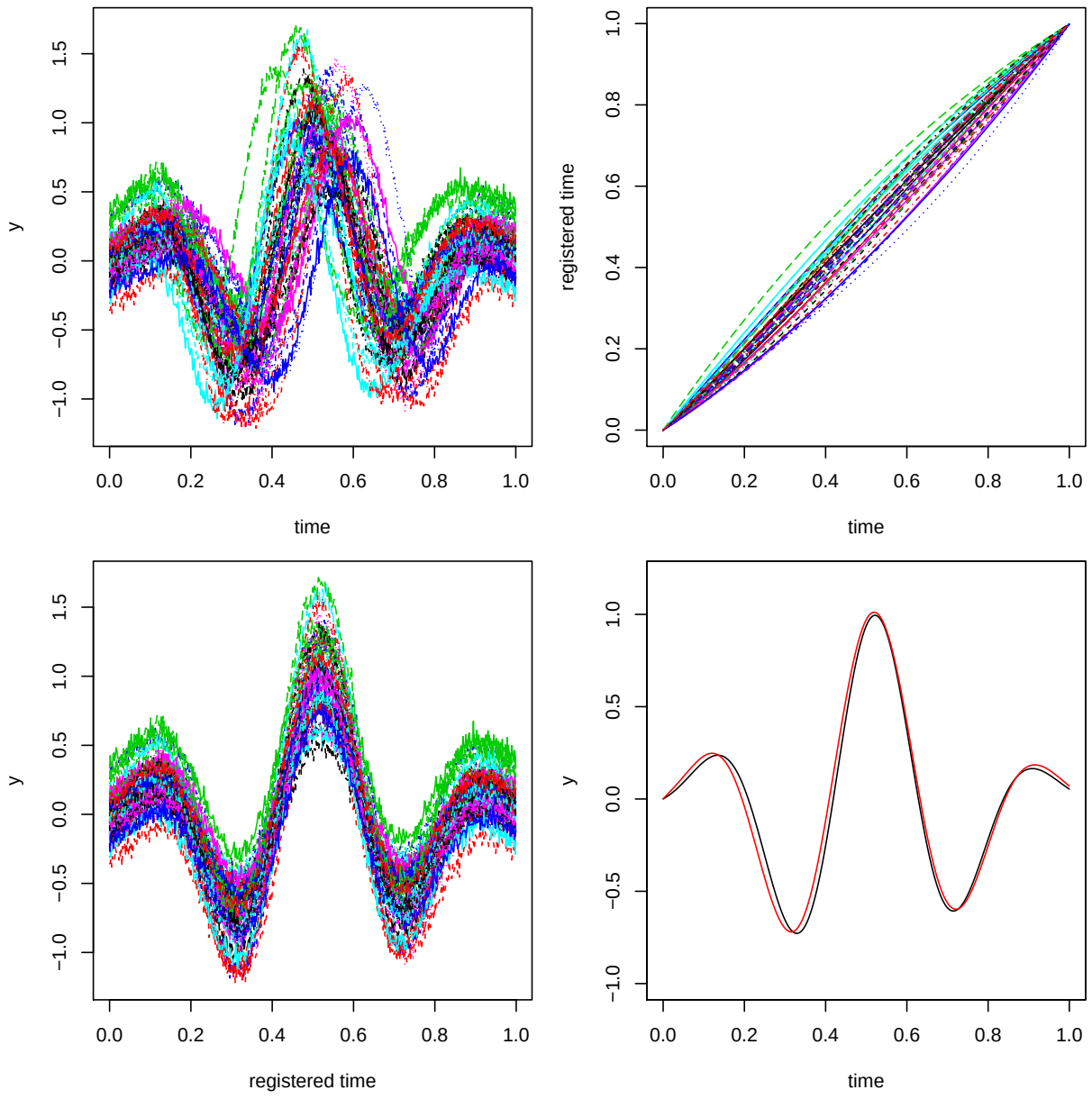

Figure 2.3: Simulation study 1.1. Upper left: unregistered functions. Upper right: original time versus registered time. Lower left: registered functions through PPM-BHCR. Lower right: Black curve is the true mean function; red curve is the cross-sectional mean of aligned functions.

As mentioned in the end of Section 2.3, we can fix $\boldsymbol{b}_i$ at its least square estimate $\hat{\boldsymbol{b}}_i$ in the simulation to save computation time. To validate this method, we compared the performance of PPM-BHCR between fixing $\boldsymbol{b}_i$ and not fixing it. We simulated 200 data sets and each of them was as above. The simulation result is in Figure 2.4, which shows no difference between these two methods. Therefore, in following study we fixed $\boldsymbol{b}_i$ at its least square estimate by default.

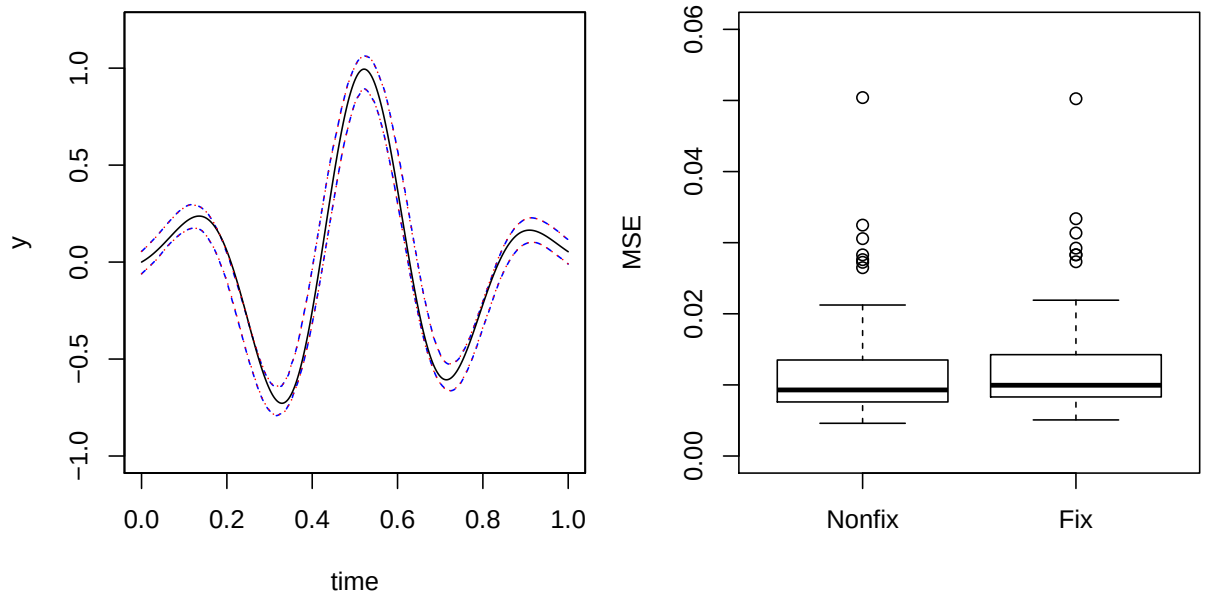

Figure 2.4: Simulation study 1.2. Left: 95% credible band of estimated mean function. Dashed blue line: estimated mean by PPM-BHCR with fixed $\boldsymbol{b}_i$. Dotted red line: estimated mean by PPM-BHCR without fixing $\boldsymbol{b}_i$. Solid black line: true mean function.

2.4.2 Simulation Study 2: Compare PPM-BHCR and Standard BHCR

To compare the predictive process model based curve registration with the standard Bayesian curve registration, we generated 200 simulation data sets. Each simulation data set was generated as Section 4.1. The two methods, PPM-BHCR and standard BHCR, were applied on these data sets respectively, both with 1000 MCMC iterations and 1000 burn-in iterations. To compare their performance, the 95% credible band for each of the estimated mean function of PPM-BHCR and standard BHCR is plotted. The MSE of difference between the estimated mean function with true mean function is also calculated for each of these two methods and showed in box plot (Figure

2.5). The simulation results show very small difference between the two methods.

In terms of time efficiency, by averaging the time cost of the 200 simulations, PPM-BHCR time cost is 36% of the time that standard BHCR costs. The larger number of time points each curve has, the more time saving PPM-BHCR achieves.

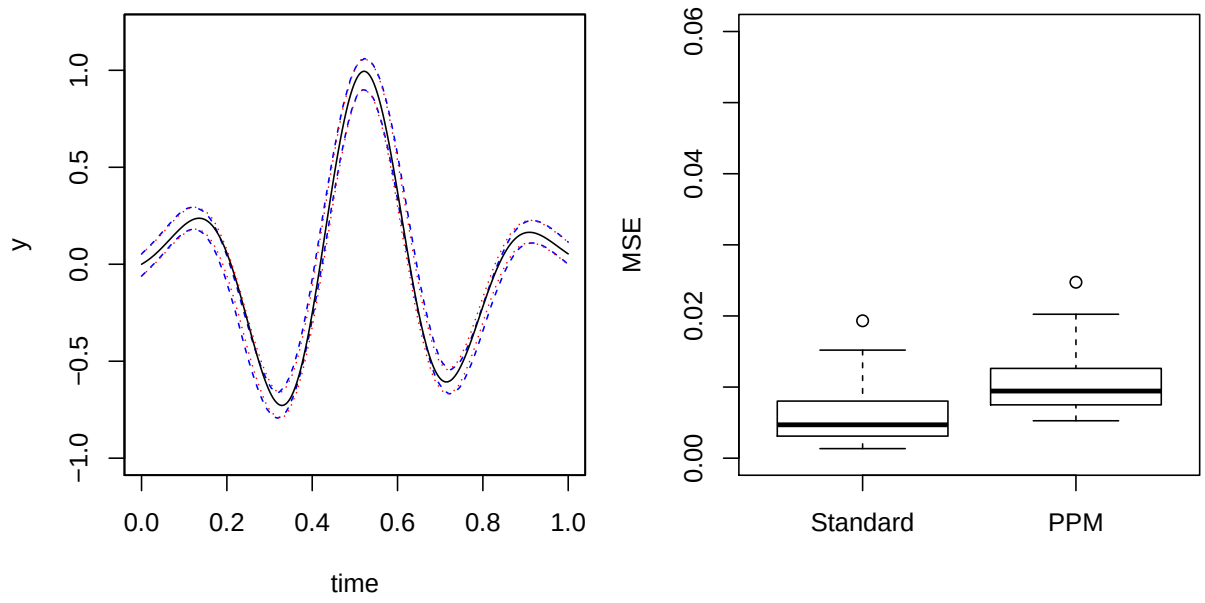

Figure 2.5: Simulation study 2. Left: 95% credible band of estimated mean function. Dashed blue line: estimated mean by standard BHCR. Dotted red line: estimated mean by PPM-BHCR. Solid black line: true mean function.

2.4.3 Simulation Study 3: Add Curves Generated by Different Function

In this simulation study we tested the robustness of PPM-BHCR when different generating functions ("noise function") exist. We generated 200 simulation data sets. In each set, we simulated $N = 45$ curves in the same way as Simulation Study 1 did, using the generating function $m(t) = sin(15t)exp(-10(t-0.5)^2)$. Then we simulated another $N = 5$ curves generated by a different function $m(t) = cos(15t)exp(-10(t-0.5)^2)$. Together, in each simulated data set there were 50 curves where 5 of them were generated by "noise function". PPM-BHCR and standard BHCR were applied to such data and their performance were compared in Figure 2.6 The simulation results show even with 10% curves generated by noise functions, PPM-BHCR still performs

very well and close to standard BHCR.

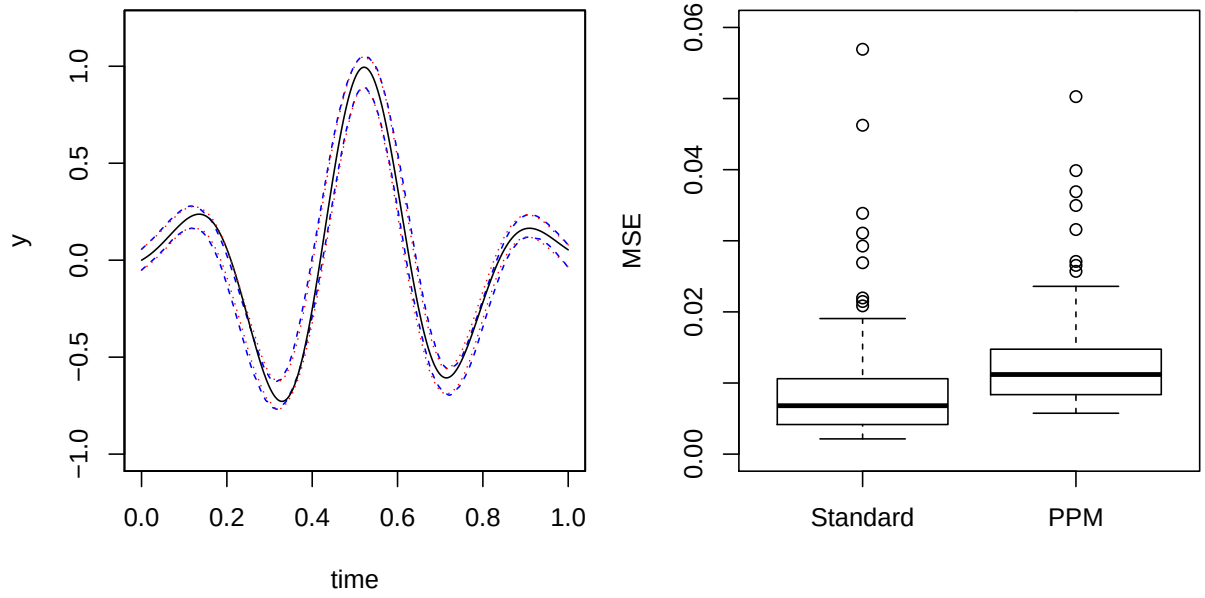

Figure 2.6: Simulation study 3. Left: 95% credible band of estimated mean function. Dashed blue line: estimated mean by standard BHCR. Dotted red line: estimated mean by PPM-BHCR. Solid black line: true mean function.

2.4.4 Simulation Study 4: Fixed b_i with Different Choices of g

In this simulation study we tested how the value of g in the prior of $\boldsymbol{b}_i$ affected model fitting results. We set $g = 0.01, 1, 100, 10000$ and run the algorithm respectively.

Simulation results are displayed in Figure 2.7. With $g = 0.01, 1, 100$, they delivered almost the same registration results, meaning the registration process is robust to variable g values. However, when g increased to 10000, the registration was less than sufficient. This is as expected. Because with large g, most of the variability of the posterior distribution of $\boldsymbol{b}_i$ comes from noise ϵ_i and dominates the MCMC algorithm.

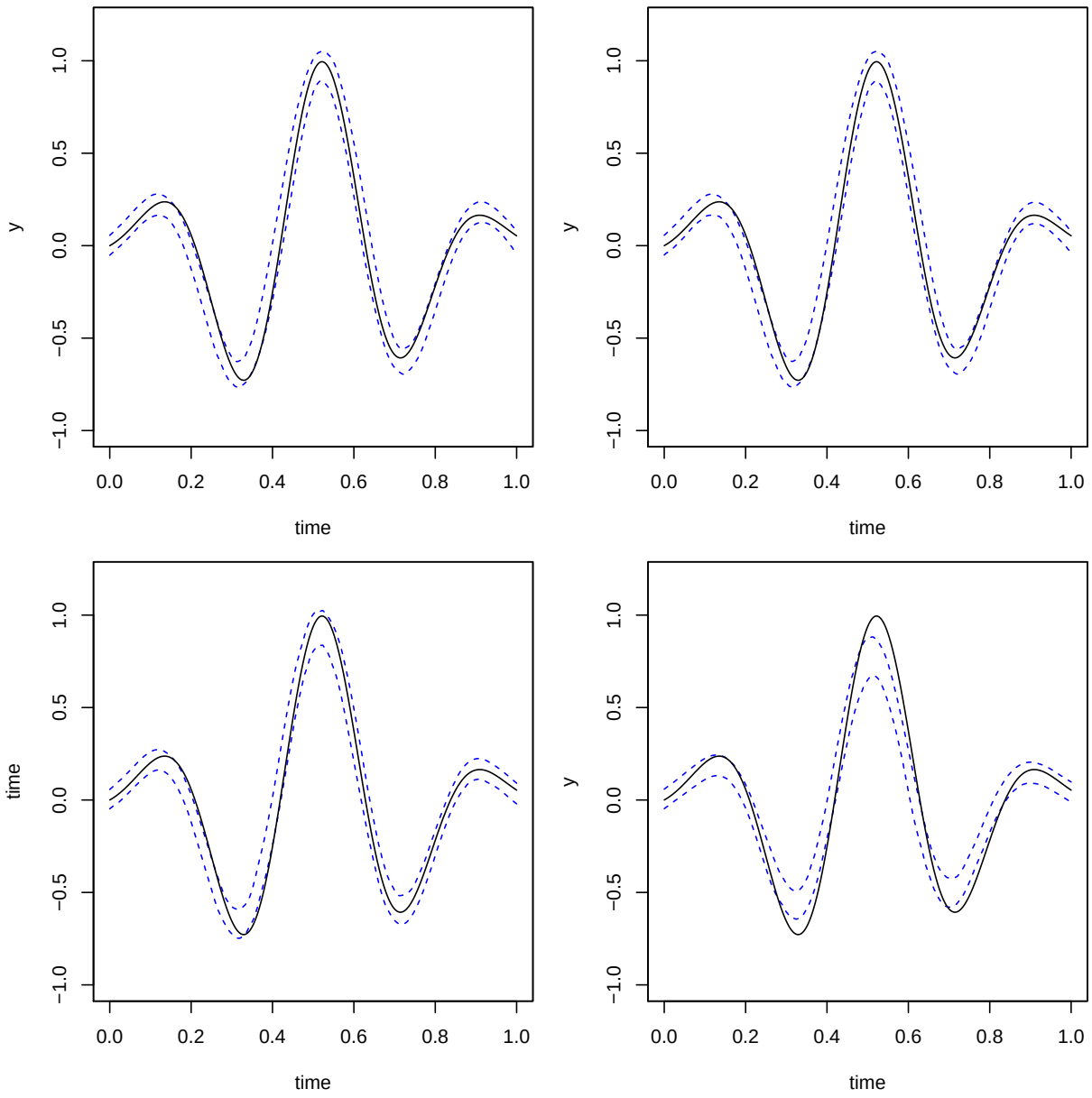

Figure 2.7: Simulation study 4. Upper left: $g = 0.01$. Upper right: $g = 1$. Lower left: $g = 100$. Lower right: $g = 10000$

2.5 Case studies

In the case study we applied PPM-BHCR method on ICP data. As described before, ICP pulses contain morphological features related to neurological activity of patients. ICP pulses also display amplitude and phase variations, which makes registration a necessary processing step.

For each patient, there are 100-200 highly intensively sampled ICP curves. The time grid is normalized to the domain $[0, 1]$. Let $k = m/5$. To estimate the time transformation function $\mu(t)$, we placed three equally spaced interior knots on $[0, 1]$. Figure 2.8 is the registration results for two patients as an example. It shows the algorithm successfully align ICP curves by their peaks and troughs. Figure 2.9 displays the the mean curves of ICP pulses for all 16 patients. Blue ICP curves are mean shape after the curve registration process, red ones are simple averaged mean function. Comparing the mean curves generated by the two different methods, for most patients they do not have significant difference.

ICP is typically triphasic ([Hu et al., 2009]), and its morphology is of clinical interest. [Eide, 2006] described wave amplitude is an important quantity to analyze ICP, which is the difference between maximum and minimum pressure value. Table 2.1 summarizes the wave amplitude in our ICP data set. With registration, amplitude is slightly higher than mean function without registration according to the quartile statistics. For individual patient, the mean function after registration always has amplitude higher than mean function without registration. Though the difference is not clinically meaningful in this case study, it can be expected if the time variation is big, registration will help avoid underestimation of amplitude.

Functional PCA is another way to explore functional data. For the ICP case, Figure 2.10 shows the first two functional principle components of mean ICP curves with or without registration account for more than 90% total variation. It implies the most variable region along time axis is from 40% to 80% of the time interval. Comparing between functional components with and without registration, the first component are very similar except the most variable region has slight phase difference. With registration, the most variable region comes around $t = 0.6$, whereas without registration the most variable region is around $t = 0.4$. For the second component, with registration it shows the region around $t = 0.6$ is more variable than without registration. In

sum, for this case study registration helps recover some low variance feature which is lost without registration.

Table 2.1: Summary Statistics of Mean Wave Amplitude.(in mmHg)

	Min	1st Quartile	Median	3rd Quartile	Max
Registered	0.66	4.12	4.94	5.95	12.45
Unregistered	0.55	4.05	4.81	5.8	11.66

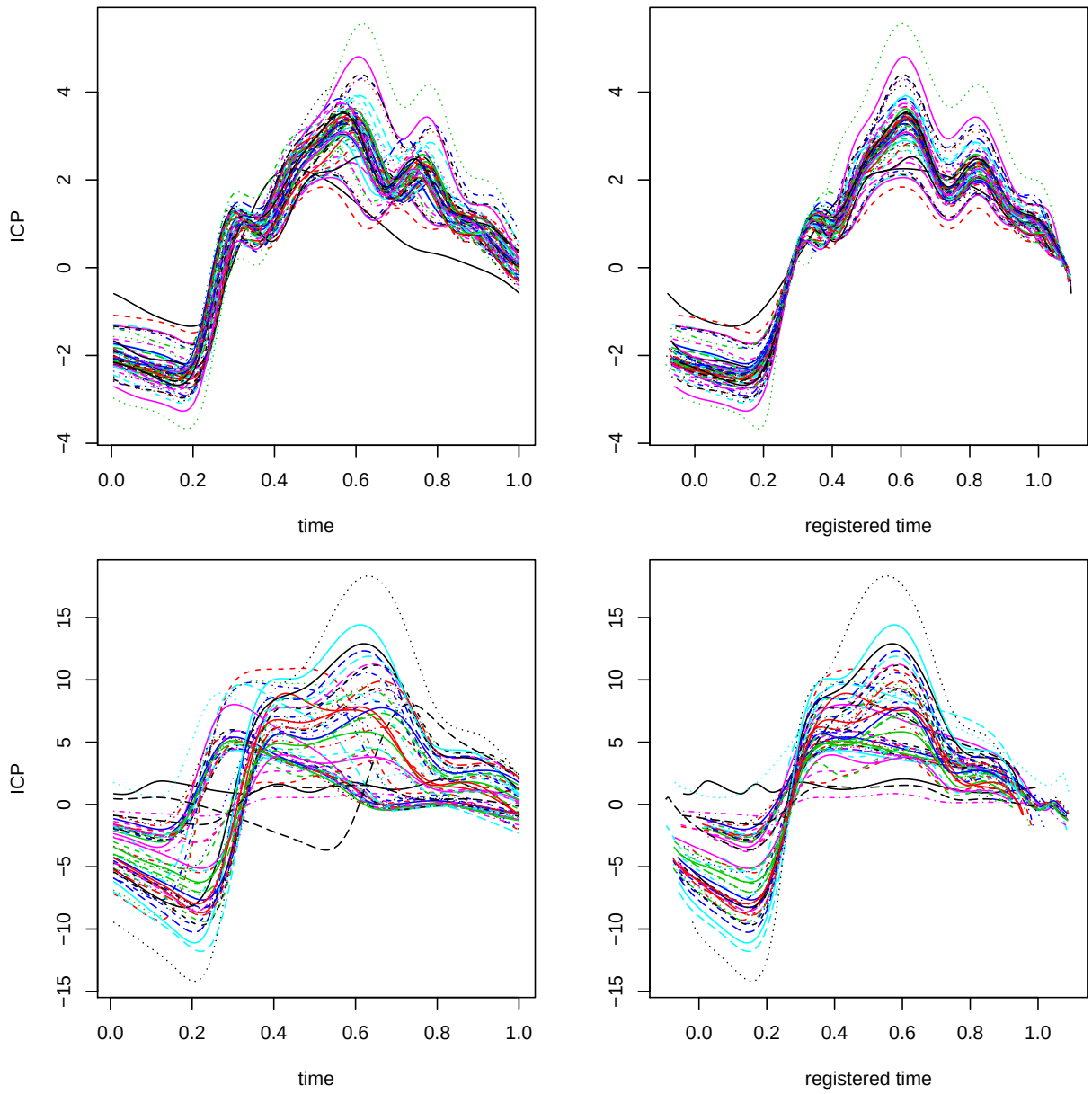

Figure 2.8: Case Study. Upper: unregistered and registered ICP curves from patient 4. Lower: Upper: unregistered and registered ICP curves from patient 14.

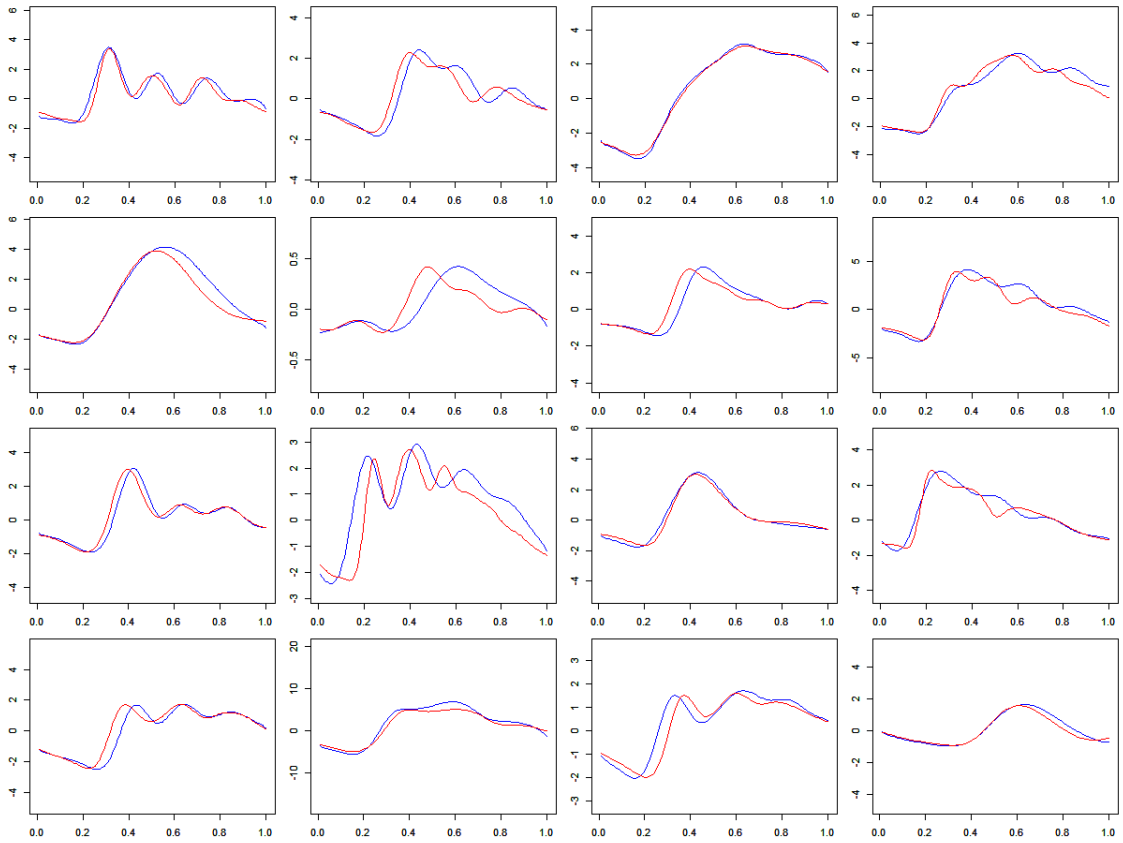

Figure 2.9: Case study. Mean ICP shape function with and without curve registration. Red curve: mean function without registration. Blue curve: mean function with registration.

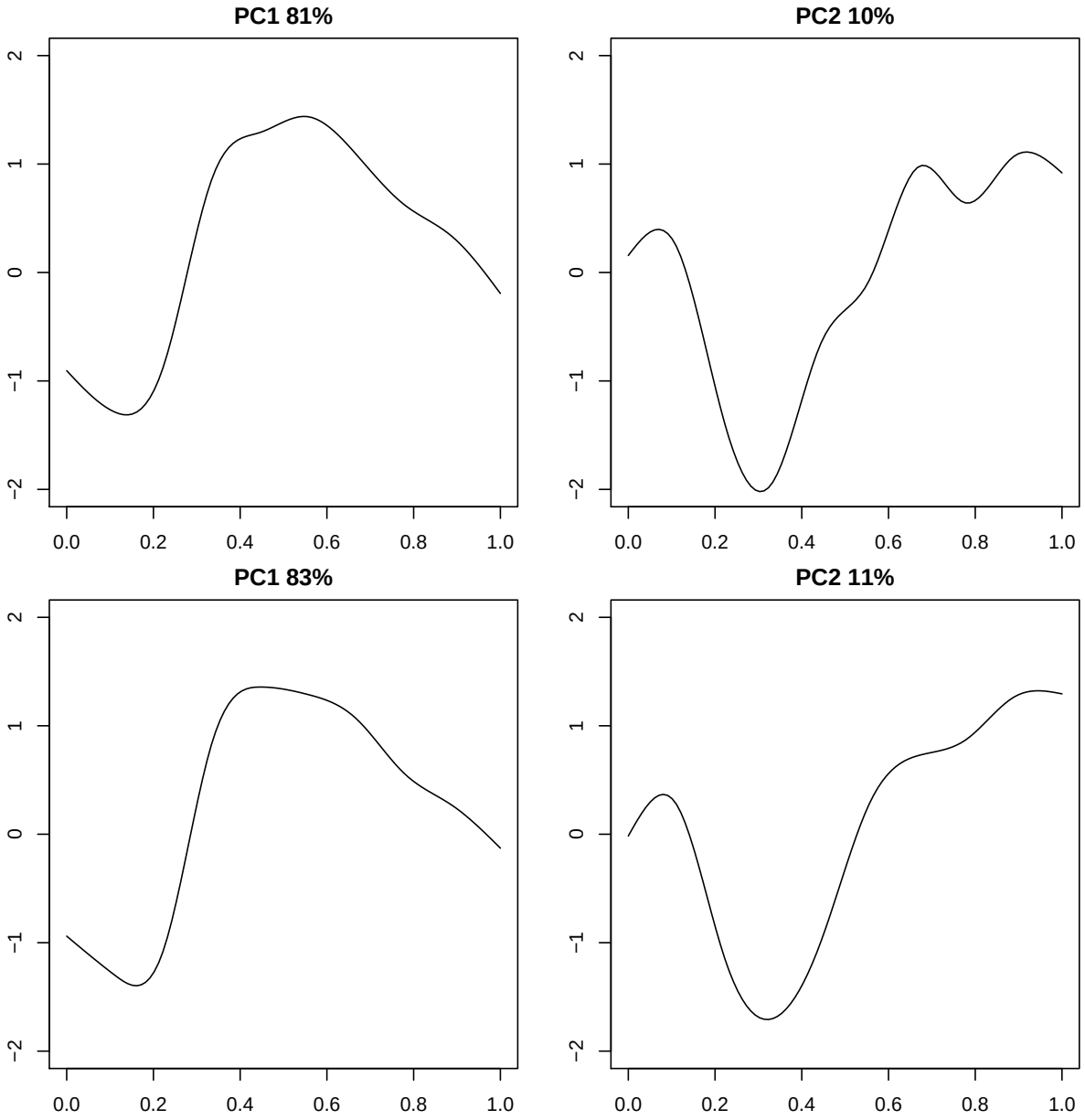

Figure 2.10: Case study. The first two functional principle components.Upper two plots: PC from mean functions after registration; Lower two plots: PC from mean functions without registration.

2.6 Discussion

In this paper we introduced a novel accelerated curve registration method when the curve is intensively sampled. There are several consideration when applying this method. First, the choice of k. The smaller k/m is, the faster the algorithm runs. How to choose k/m is a question equivalent to the choice of number of knots in smoothing spline problem. A good balance between speed and accuracy comes from practice. In our simulation study, letting $k/m = 5$ leads to satisfactory result. Second, the choice of g. Too large g will lead to unsatisfactory alignment, as shown in our simulation study. Finally, when comparing PPM-BHCR with standard BHCR, the latter one still has lower MSE, however much slower speed. In the case study of aligning ICP curves, PPM-BHCR successfully obtained synchronized ICP curves, and provided mean ICP curve after the alignment. Figure 2.8 shows example from two patients, where the original ICP curves have phase variability, and the PPM-BHCR method aligns the curves. Figure 2.9 shows the mean ICP curve for each patient after the ICP curves are aligned. With these mean curves, we are able to draw summary statistics upon them for further analysis. We can also plot the functional PCA components to examine where the most variability comes from. In the ICP case study, the amplitude of mean function and variable features are not very different between with and without registration. However, it can be expected if the data show big phase variation, registration will help avoid underestimation of amplitude and recover low variance features.

The ICP data used in case study is longitudinal-functional, since ICP pulses of the same patient were measured sequentially. To model such data, the correlation between ICP curves within a subject needs to be considered. In the current implementation, this correlation is captured through amplitude and scale parameter c_i and a_i. It does not take the sequential order into consideration. Ideally, it is expected ICP pulses measured closer to each other have stronger correlation than pulses that are remote from each other. To model such correlation, a random effect function $U_i(t)$ can be added, where $U_i(t)$ is mean zero Gaussian process with covariance $Q(t_h, t_l)$ representing the within-function covariance structure.

To tackle the computing speed problem of curve registration, [Earls and Hooker, 2015] proposed adapted variational Bayes procedure for approximate inference as described in introduction.

Comparing the variational Bayes method with our predictive process method, they are fundamentally different. Variational Bayes method relies on the optimization of approximating posterior distribution, which can be very difficult depending on how complex it is. On the other hand, variational Bayes method can be much faster than predictive process method, since if the optimization goes well, the algorithm converges within a dozen of rounds. Predictive process method is more empirical. Its speed and accuracy are controlled by the k/m ratio. It works no matter how complex the curve shape is. However, it may not provide satisfactory result if white noises are large in data.

2.7 Appendix: Full Conditionals

In this appendix we present the full conditionals used for PPM-BHCR. To simplify notation, we omit t from formulation, use S_i denote the design matrix $S_m(\mu_i)$, and use $\boldsymbol{m}_i$ denote $S_m(\mu_i)\boldsymbol{\beta}$.

- Full conditional distribution for $\boldsymbol{b}_i$ $(\boldsymbol{b}_i|\boldsymbol{Y}_i, \boldsymbol{\theta}_{-b_i}) \sim N(\boldsymbol{b}_i^*; V_{b_i})$, where
  $V_{b_i} = (\frac{1}{\sigma_\epsilon^2} S_f' S_f + \frac{1}{g\sigma_\epsilon^2} I_k)^{-1}$, $\boldsymbol{b}_i^* = V_{b_i}(\frac{1}{\sigma_\epsilon^2} S_f' y_i + \frac{1}{g\sigma_\epsilon^2}\boldsymbol{\gamma}_i)$
- Full conditional distribution for $\boldsymbol{\beta}$ $(\boldsymbol{\beta}|\boldsymbol{b}, \boldsymbol{\theta}_{-\beta}) \sim N(\boldsymbol{\beta}^*; V_\beta)$, where
  $V_\beta = (\frac{1}{g\sigma_\epsilon^2}\sum a_i^2 S_i' S_i + \frac{\Omega_\beta}{\lambda_\beta})^{-1}$, $\boldsymbol{\beta}^* = V_\beta \frac{1}{g\sigma_\epsilon^2}\sum a_i S_i'(\boldsymbol{b}_i - c_i \mathbf{1}_k)$
- Full conditional distribution for c_i $(c_i|\boldsymbol{b}_i, \boldsymbol{\theta}_{-c_i}) \sim N(c_i^*; V_{c_i})$, where
  $V_{c_i} = (\frac{k}{g\sigma_\epsilon^2} + \frac{1}{\sigma_c^2})^{-1}$, $c_i^* = V_{c_i}\frac{1}{g\sigma_\epsilon^2}\mathbf{1}_k'(\boldsymbol{b}_i - a_i\boldsymbol{m}_i)$
- Full conditional distribution for a_i $(a_i|\boldsymbol{b}_i, \boldsymbol{\theta}_{-a_i}) \sim N(a_i^*; V_{a_i})$, where
  $V_{a_i} = (\frac{1}{g\sigma_\epsilon^2}\boldsymbol{m}_i'\boldsymbol{m}_i + \frac{1}{\sigma_a^2})^{-1}$, $a_i^* = V_{a_i}\frac{1}{g\sigma_\epsilon^2}(\boldsymbol{b}_i - c_i\mathbf{1})'\boldsymbol{m}_i$
- Full conditional distribution for σ_ϵ^2 $(\sigma_\epsilon^2|\boldsymbol{Y}, \boldsymbol{\theta}_{-\sigma_\epsilon^2}) \sim IG(a_\epsilon^*, b_\epsilon^*)$
  $a_\epsilon^* = a_\epsilon + mn/2$, $b_\epsilon^* = b_\epsilon + \sum(\boldsymbol{Y}_i - S_f\boldsymbol{b}_i)'(\boldsymbol{Y}_i - S_f\boldsymbol{b}_i)$
- Full conditional distribution for σ_c^2 $(\sigma_c^2|\boldsymbol{Y}, \boldsymbol{\theta}_{-\sigma_c^2}) \sim IG(a_c^*, b_c^*)$
  $a_c^* = a_c + n/2$, $b_c^* = b_c + 1/2\sum c_i^2$

- Full conditional distribution for σ_a^2 $(\sigma_a^2|\boldsymbol{Y}, \boldsymbol{\theta}_{-\sigma_a^2}) \sim IG(a_a^*, b_a^*)$

  $a_a^* = a_a + n/2$, $b_a^* = b_a + 1/2\sum(a_i - 1)^2$

- Full conditional distribution for λ_β $(\lambda_\beta|\boldsymbol{Y}, \boldsymbol{\theta}_{-\lambda_\beta}) \sim IG(a_\beta^*, b_\beta^*)$

  $a_\beta^* = a_\beta + p/2$, $b_\beta^* = b_\beta + 1/2\boldsymbol{\beta}'\Omega_\beta\boldsymbol{\beta}$, p is the length of $\boldsymbol{\beta}$.

- Full conditional distribution for λ_ϕ $(\lambda_\phi|\boldsymbol{Y}, \boldsymbol{\theta}_{-\lambda_\phi}) \sim IG(a_\phi^*, b_\phi^*)$

  $a_\phi^* = a_\phi + nw/2$, $b_\phi^* = b_\phi + 1/2\sum(\boldsymbol{\phi}_i - \boldsymbol{\Upsilon})'\Omega_\phi(\boldsymbol{\phi}_i - \boldsymbol{\Upsilon})$, w is the length of $\boldsymbol{\phi}_i$.

CHAPTER 3

Bayesian Warped Functional Regression

3.1 Introduction

Functional regression can model curves as functions of other curves. For example, we have a motivating data set of daily trajectories of oxides of nitrogen in the city of Sacramento, California, on 52 summer days in the year 2005, and the corresponding trajectories of ozone concentration. The goal is to predict ozone concentration from the concentration of oxides of nitrogen. Many literatures and work have been accomplished to tackle such research topic. However, most of them focus on modeling the characteristics of curve amplitude using functional principle components. The phase variability is not widely investigated under regression setting. When phase variability presents in data, a large number of principle components may be needed in order to provide good fit to data. Such strategy is neither efficient nor easy to interpret. A better strategy is to use functional principle components to model amplitude, and use warping model to model phase variability, and combine these two models together. [Gervini, 2015] proposed a joint model which incorporates registration as an intrinsic part of the regression model for the case of historical function-on-function regression. In addition to efficiency and easy-to-interpret, such model is also able to predict new un-synchronized response curve. The inference is done in frequentist's fashion. In this chapter, we present a Bayesian framework for integrating curve registration and regression under one model, called Bayesian Warped Functional Regression (BWFR). We consider two types of function-on-function regression: historical and concurrent functional regression. The functional coefficient is decomposed through functional principle components, and parameter estimation is done by MCMC sampling. Simulation study is conducted to evaluate the performance of our proposed method. The regression model is also applied to two motivating case studies. Both

case studies have functional response and predictor which display amplitude and phase variability. One case study is lip movement data, where the goal is to explore functional relationship between neural activity in lip muscle and lip movement when speaking words. The other case study is the air pollution data of Sacramento as aforementioned, where the goal is to explore the relationship between oxides of nitrogen and ozone concentration.

3.2 Model Formulation

Consider functional predictor and response denoted by (x_i, y_i), $i = 1, ..., n$. Their mean functions are $Ex(t) = \mu_x(t)$ and $Ey(t) = \mu_y(t)$ respectively. Their covariance functions are $cov(x(s), x(t)) = G_x(s,t)$ and $cov(y(s), y(t)) = G_y(s,t)$ respectively. By Karhunen-Loéve expansion, these covariance function can be expanded by orthogonal eigenfunctions: $G_x(s_1, s_2) = \sum \rho_k \phi_k(s_1)\phi_k(s_2)$, and $G_y(t_1, t_2) = \sum \lambda_l \psi_l(t_1)\psi_l(t_2)$ with eigenvalues ρ_k and λ_l. Then the observed predictor and response are:

$$U_{iu} = x_i(s_{iu}) + \epsilon_{iu} = \mu_x(s_{iu}) + \sum_k \zeta_{ik}\phi_k(s_{iu}) + \epsilon_{iu} \tag{3.1}$$

$$V_{iv} = y_i(t_{iv}) + \varepsilon_{iv} = \mu_y(t_{iv}) + \sum_l \xi_{il}\psi_l(t_{iv}) + \varepsilon_{iv} \tag{3.2}$$

The errors ϵ_{iu} and ε_{iv} are iid white noise. ζ_{ik} and ξ_{il} are functional principle component scores.

A historical functional regression model is:

$$E[y(t)|x] = \alpha(t) + \int_0^t \beta(s,t)x(s)ds$$

Assuming x and y are square-integrable functions on a common interval $[0, 1]$. Let $x^c(t) = x(s) - \mu_x(s)$, and given that $Ey(t) = \mu_y(t) = \alpha(t) + \int_0^t \beta(s,t)\mu_x(s)ds$, the above model becomes:

$$E[y(t)|x] = \mu_y(t) + \int_0^t \beta(s,t)x^c(s)ds \tag{3.3}$$

Such functional regression is called historical because it is clear that future values of function x has no impact on y.

To estimate the regression function, $\beta(s,t)$ is decomposed as in [Müller et al., 2008]:

$$\beta(s,t) = \sum_k^p \sum_l^q b_{kl}\phi_k(s)\psi_l(t) \tag{3.4}$$

where $\phi_k(s)$ and $\psi_l(t)$ are orthogonal eigenfunctions as introduced before. Such decomposition avoids identifiability issues since it can be shown ([He et al., 2000]):

$$\beta(s,t) = \sum_{k}^{\infty}\sum_{l}^{\infty} \frac{E[\zeta_k \xi_l]}{E[\xi_l^2]}\phi_k(s)\psi_l(t) \tag{3.5}$$

A special case of historical model is called concurrent model ([Ramsay and Silverman, 2005]), where only current value of predictor has impact on response:

$$E[y(t)|x] = \beta_0(t) + \beta_1(t)x(t) \tag{3.6}$$

In such model, $\beta_0(t)$ and $\beta_1(t)$ are usually decomposed by B-splines ([Huang et al., 2004]): $\beta(t) = \sum_r \gamma_r B_r(t)$.

Model 3.3 and 3.6 are ordinary functional regression, and they work just fine for synchronized functional observations. Suppose we have a common warping function $w_i(t)$ underlying misaligned observed $x_i(t)$ and $y_i(t)$. To simplify notation, let $x_i(t)$ denote centered $x_i^c(t)$. Applied model 3.3 to synchronized curves $\tilde{x}_i(t) = x_i(w_i^{-1}(t))$ and $\tilde{y}_i(t) = y_i(w_i^{-1}(t))$:

$$\tilde{y}_i(t) = \mu_y(w_i^{-1}(t)) + \int_0^{w_i^{-1}(t)} \beta(s,t)\tilde{x}_i(s)ds + \epsilon_i \tag{3.7}$$

Let $\mu_i(t) = w_i^{-1}(t)$. In this model we want to estimate time transformation function $\mu_i(t)$ which synchronize curves, and functional slope $\beta(s,t)$ simultaneously. Let $\phi_k(s)$ and $\psi_l(t)$ be eigenfunctions of covariance function of $\tilde{x}$ and $\tilde{y}$ respectively. The integration part can be further derived as:

$$\int_0^t \beta(s,t)\tilde{x}_i(s)ds = \Big\{ \int_0^t \boldsymbol{\phi}^T(s)\tilde{x}_i(s)ds \Big\}\mathbf{B}_b\boldsymbol{\psi}(t) \tag{3.8}$$

where $\mathbf{B}_b$ is matrix of b_{kl}. Let $\gamma_i(t) = \int_0^t \boldsymbol{\phi}^T(s)\tilde{x}_i(s)ds$, then we have:

$$\int_0^t \beta(s,t)\tilde{x}_i(s)ds = \Big\{\boldsymbol{\psi}(t)^T \otimes \boldsymbol{\gamma}_i(t)^T\Big\} vec(\mathbf{B}_b) \tag{3.9}$$

so that the estimation of $\beta(s,t)$ is obtained through estimating vectorized $\mathbf{B}_b$.

For estimating time transformation function $\mu_i(t)$, we use B-spline basis and Jupp transformation ([Jupp, 1978]). Time transformation function $\mu_i(t)$ needs to be strictly monotonically increasing and confined on the support of t: $t_1 \le \mu_i(t_2) < \mu_i(t_3) < ... < \mu_i(t_{m-1}) \le t_m$. Let $\mu_i(t)$ be decomposed by B-spline:

$$\mu_i(t) = \mathbf{B}'_\mu(t)\boldsymbol{\phi}_i \tag{3.10}$$

with constraint on the components of $\boldsymbol{\phi}_i$: $t_1 = \varphi_{i1} < \varphi_{i2} < ... < \varphi_{iQ} = t_m$ to ensure the time transformation is monotonic increasing. Define Jupp transformation as $Jupp(\boldsymbol{\phi}_i) = \boldsymbol{\tau}_i$, where $\boldsymbol{\tau}_i = (\tau_{i1}, ..., \tau_{iQ})$:

$$\tau_{ij} = \begin{cases} log\{(\varphi_{i,j+1} - \varphi_{ij})/(\varphi_{ij} - \varphi_{i,j-1})\}, & \text{if } j = 2, ..., Q-1 \\ \varphi_{ij}, & \text{if } j = 1, Q. \end{cases}$$

τ_{ij}s are unconstrained parameters and easier to draw sample from their posterior distribution. The reverse Jupp transformation is defined as $\boldsymbol{\phi}_i = Jupp^{-1}(\boldsymbol{\tau}_i)$, which is a vector of increasing elements with $\boldsymbol{\phi}_{i1} = \tau_{i1}$ and $\boldsymbol{\phi}_{iQ} = \tau_{iQ}$.

We use Gaussian prior for vectorized $\mathbf{B}_b$ and unconstrained parameter $\boldsymbol{\tau}_i$:

$$vec\mathbf{B}_b \sim N(\mathbf{0}, \Lambda_B/g) \tag{3.11}$$

where Λ_B is Kronecker product of eigenvalues λ_ϕ and λ_ψ of the covariance matrices of x and y respectively.

$$\boldsymbol{\tau}_i \sim N(\boldsymbol{\Upsilon}_J, \sigma_\tau^2 \mathbf{I}) \tag{3.12}$$

where $\boldsymbol{\Upsilon}_J$ is the Jupp transformation of the identity transformation function which satisfies $\mu(\tau; \boldsymbol{\Upsilon}) = \tau$. We also assign Inverse Gamma prior for hyperparameter σ_ϵ^2 and σ_τ^2.

For concurrent functional regression model, the same normal prior with a first order random walk shrinkage is assigned on the functional coefficient $\boldsymbol{\gamma}_0$ and $\boldsymbol{\gamma}_1$:

$$\boldsymbol{\gamma} \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\gamma) \tag{3.13}$$

This is similar as the coefficient for shape function in BHCR, where the covariance matrix $\boldsymbol{\Sigma}_\gamma$ is a special banded matrix.

3.3 Estimation

Model parameter estimation is done by MCMC simulation. Based on model 3.7, the regression is implemented on synchronized curves $\tilde{x}$ and $\tilde{y}$. This is realized by first estimating the time transformation function $\mu_i(t)$ for each subject, and then compute $\tilde{x}$ and $\tilde{y}$ via spline interpolation. The algorithm is described as following:

1. For $i = 1, ..., n$, simulate each $\boldsymbol{\tau}_i$ from their full conditional posterior via Metropolis-Hastings algorithm. More specifically, during each iteration, the vector $\boldsymbol{\tau}_i$ is proposed from a multivariate normal distribution $N(\boldsymbol{\tau}_0, V)$. $\boldsymbol{\tau}_0$ corresponds to φ that makes identity time transformation. V is a matrix subject to adjustment during simulation to achieve ideal acceptance rate.

2. For $i = 1, ..., n$, do reverse Jupp transformation to compute $\mu_i(t)$ based on simulated $\boldsymbol{\tau}_i$.

3. For $i = 1, ..., n$, compute warped response and predictor: $\tilde{y}_i(t) = y_i(\mu_i(t))$ and $\tilde{x}_i(t) = x_i(\mu_i(t))$ by spline interpolation.

4. Re-compute functional PCA basis $\phi(t)$ and $\psi(t)$ based on the warped response and predictor $\tilde{y}$ and $\tilde{x}$

5. Simulate $\mathbf{B}_b$, σ_ϵ and σ_τ^2 from their full conditional posterior via Gibbs sampling method.

After obtaining the simulated samples, estimation for each parameter is calculated as the average over these simulated samples. Additional statistical inference can also be drawn based on these samples.

3.4 Simulation

3.4.1 Simulation 1

In this simulation, we examine basic performance of the proposed method for both historical and concurrent functional regression model. For historical model, we simulate a data set with $n = 30$ subjects. For each subject, $y(t)$ and $x(t)$ are observed on the same time grid $t = [0, 1]$, with $m = 50$ equally spaced time points. Functional coefficient is generated from $\beta(s, t) = exp(-50(s - 0.4)^2 - 20(t - 0.6)^2)$. The effect of this regression function β is to shift the peak in x from 0.4 to 0.6 in y. Warping function is simulated by Beta density function $w_i(t) = Beta(a_1, a_2)$ where a_1, a_2 are from reversed Gamma distribution. Mis-aligned functional predictor is generated from $x_i(s) \circ \mu_i(s) = z_i * exp(-30 * (s - 0.4)^2) \circ w_i(s)$ where $z_i \sim N(1, 0.2)$. Functional outcome

is generated from $y_i(t) \circ w_i(t) = \beta(t)x_i(w_i(t)) \circ w_i(t) + \epsilon_i$ where $\epsilon_i \sim N(0, 0.1)$. The MCMC simulation has 1500 samples with the first 500 as burn-in.

Figure 3.1 displays the unwarped and warped outcome y and x. In the upper panel, the phase variabilities are quite obvious in both x and y. The algorithm successfully synchronized them as shown in the middle panel. The lower panel shows the fitted response variable.

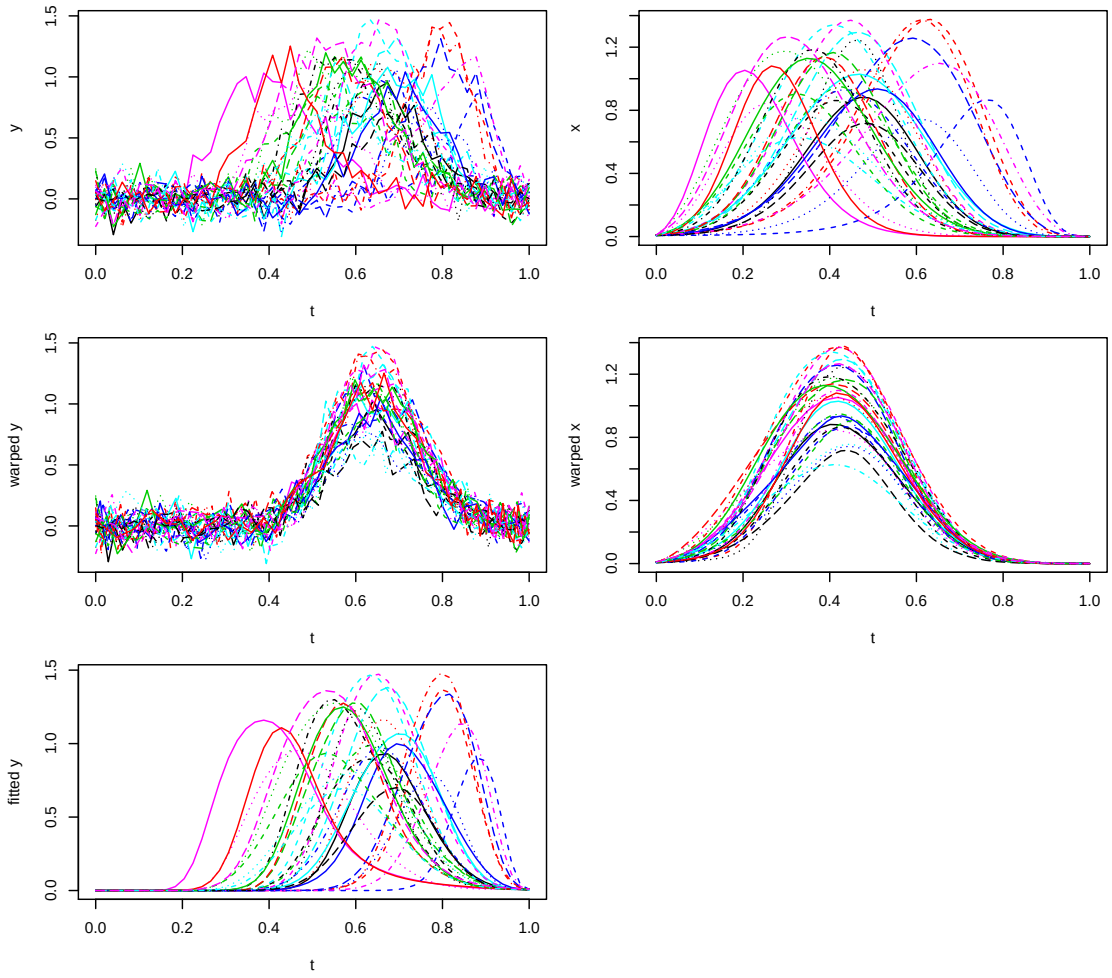

Figure 3.1: Simulation study 1.1 historical model: upper panel: unwarped functional observations; middle panel: warped functional observations; lower panel: fitted response variable.

For concurrent model, x is simulated from $exp(-20 * (s - 0.5)^2)$, and regression function is $\beta(t) = exp(-20 * (t - 0.8)^2 - 20 * (t - 0.6)^2)$. The other settings are the same as historical model. Figure 3.2 displays the unwarped and warped outcome y and x. The algorithm successfully aligned response and predictor variable just like historical model.

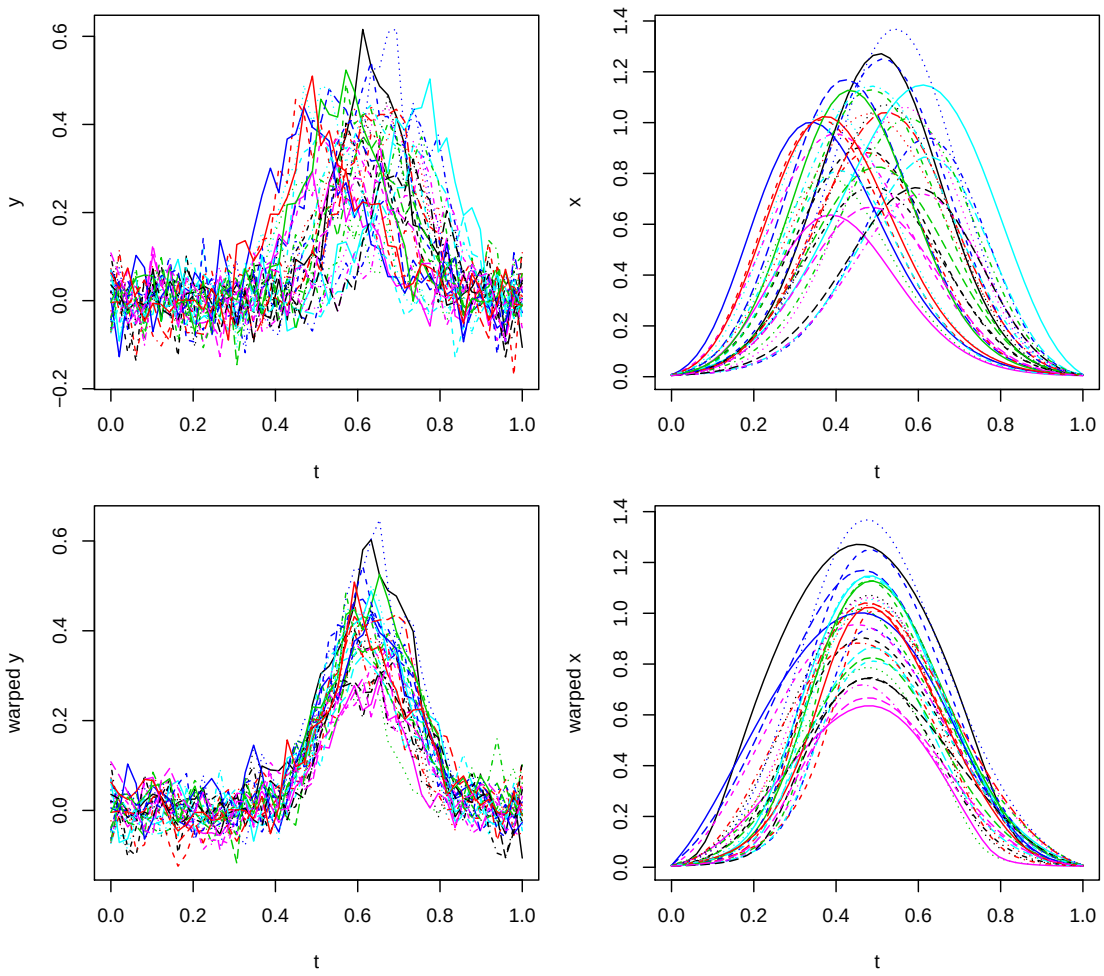

Figure 3.2: Simulation study 1.1 concurrent model: upper panel: unwarped functional observations; lower panel: warped functional observations.

Figure 3.3 displays the estimated $\beta(s,t)$ compared to the real one. The first thing we can tell from this figure is that the estimation by functional regression without warping is not acceptable. From this contour plot we can tell that the regression function put heavy weight on predictor around $s = 0.4$ and response around $t = 0.6$, making these two regions in predictor and response positively related. Figure 3.4 shows the confidence band of $\beta(s,t)$ at a specific time point. As described before, the impact that regression function $\beta(s,t)$ makes is to move the peak in x from $s = 0.4$ to $t = 0.6$ in y. Therefore, it is of interest to see how the estimation performs at $t = 0.6$. The regression function at this time point $\beta(s, t = 0.6)$ can be imagined as a slice from the 3-D surface of $\beta(s,t)$. It shows at the time when y peaks, the regression function puts most weight on

x at $s = 0.4$. Additionally, the confidence band shows the estimation of $\beta(s,t)$ at $t = 0.6$ still has some room to improve.

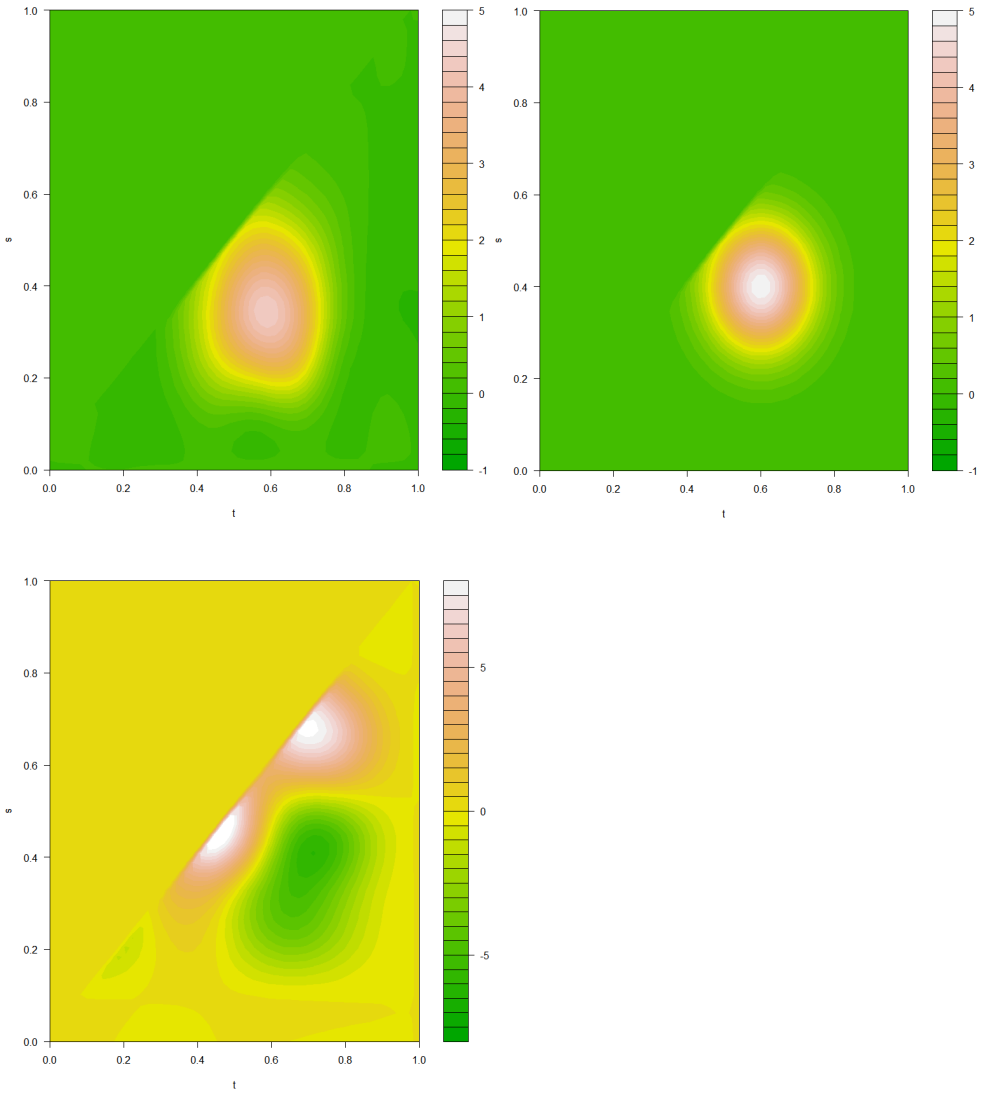

Figure 3.3: Simulation study 1.1 Contour plot of $\beta(s,t)$: upper left: estimated $\beta(s,t)$ by historical functional regression with warping; upper right: real $\beta(s,t)$; lower left: estimated $\beta(s,t)$ by historical functional regression without warping.

Next we evaluate the prediction accuracy. We simulate 100 data sets as what we did for historical model, and fit each data set using historical warped functional regression, concurrent warped functional regression and ordinary functional regression. Figure 3.5 shows the ordinary functional regression without warping has the worst performance. Historical warped functional regression has

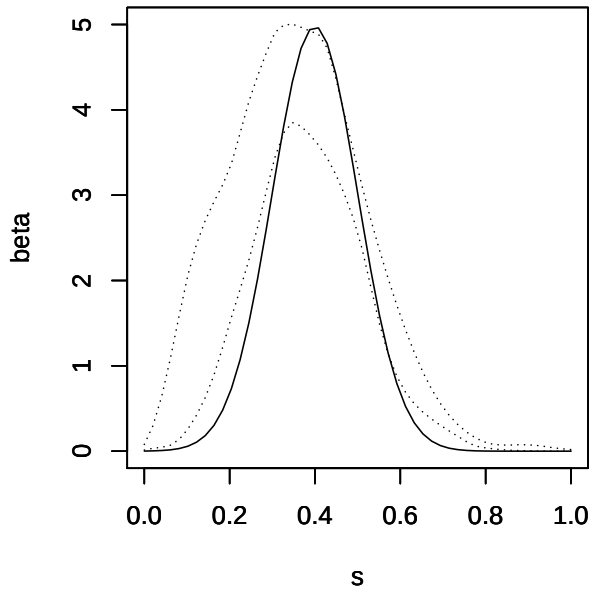

Figure 3.4: Simulation study 1.1 $\beta(s,t)$ at a specific time point $t = 0.6$; Solid line is the real value of $\beta(s,t)$, dotted line is 95% confidence band.

better performance than concurrent regression, since the model is correctly specified according to simulation structure.

3.4.2 Simulation 2

In this simulation study, we test the sensitivity of algorithm to different values of tuning parameter g. The value of g ranges from 0.01, 1, 100 and 10000. Based on a single simulated data set, Figure 3.6 shows when $g = 0.01$, the alignment of curves are not satisfactory. We then simulate 100 data sets, and plot the MSE for fitted outcome and estimated warping function, Figure 3.7 shows $g = 1$ offers the best performance.

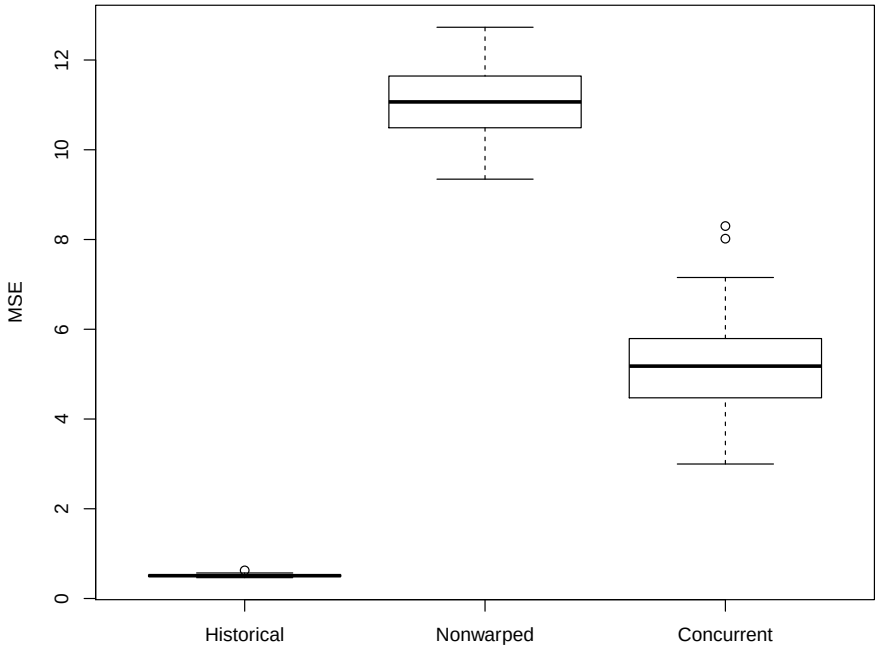

Figure 3.5: Simulation study 1.2: Boxplot of MSE for Fitted Outcome.

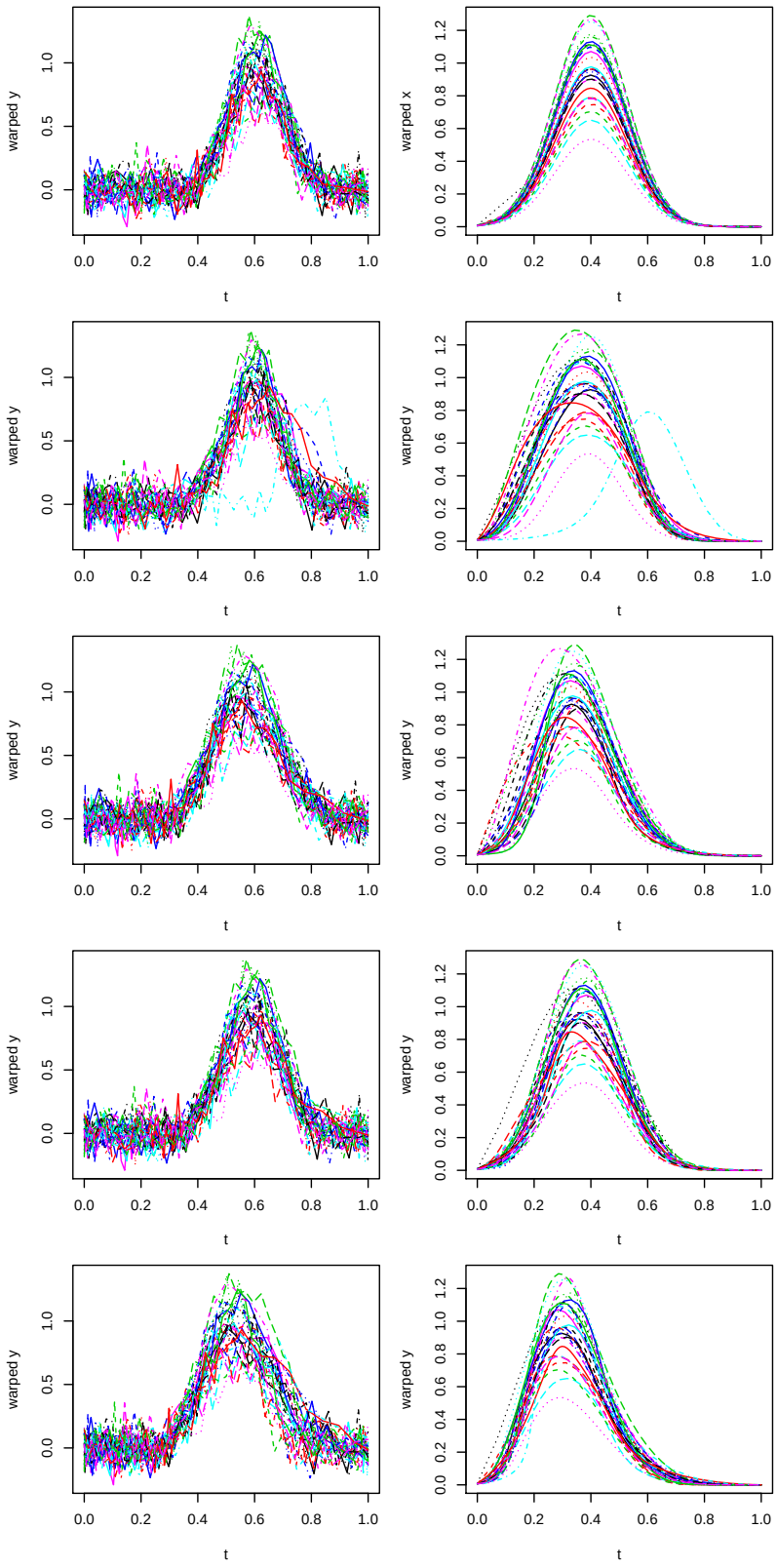

Figure 3.6: Simulation study 2. Warped functional observations by different g values.

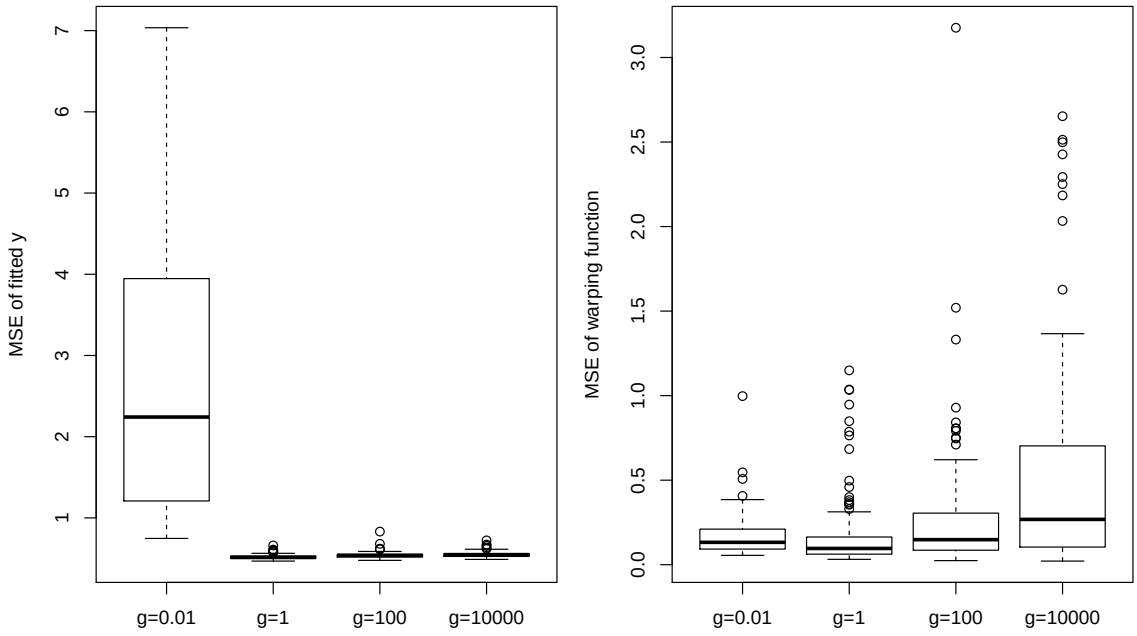

Figure 3.7: Simulation study 2. Boxplot of MSE by different g values.

3.5 Case Study

3.5.1 Case Study 1: Lip Movement

Lip movement data come from [Malfait and Ramsay, 2003]. In the experiment, a person was asked to speak "Bob" a few times. The lip movement and associated electromyography (EMG) curves were recorded. Then lip acceleration curves were obtained by differentiating the smoothed lip movement curves.

Before fit the model, we first plotted the first four eigenfunctions of predictor and response, as shown in Figure 3.8. The first four eigenfunctions account for more than 90% if the total variations. When we fit the historical model, it is sufficient to set the number of PCA basis functions $p = q = 4$.

In this case study, both historical and concurrent warped functional regression model are fitted to the lip movement data respectively. The measure of lip acceleration is considered as outcome, and the measure of neural activity (EMG) of lip muscle is used as predictor. Figure 3.9 shows the curve alignment by these two models, where one does not appear superior to the other. The MSE for fitted y is 33.67 by historical model versus 58.33 by concurrent model. We also use Watanabe-Akaike information criterion (WAIC) to compare the prediction accuracy of these two models. WAIC is considered as an improvement to deviance information criterion (DIC) for Bayesian models. WAIC uses the entire posterior distribution, and it is asymptotically equal to cross-validation. Moreover, WAIC is invariant to parametrization. WAIC is calculated by:

$$WAIC = \sum_i V_s(log[p(y_i|\boldsymbol{\theta}^s)]) \tag{3.14}$$

where V_s is the sample variance calculator, $p(y_i|\boldsymbol{\theta}^s)$ is likelihood. Applying WAIC to the above two models, WAIC for historical model is 2819 and for concurrent model is 5180. Therefore historical model is a better fit to this data set. The contour plot Figure 3.10 can help interpret the regression function.

To interpret $\beta(s,t)$, we first fix t at a specific time point of interest, for example, at $t = 0.6$, where we see strong relationship from the contour plot. Then we look at Figure 3.11. Around $s = 0.2$ and $s = 0.45$, the predictor is strongly negatively related to response, and around $s = 0.3$

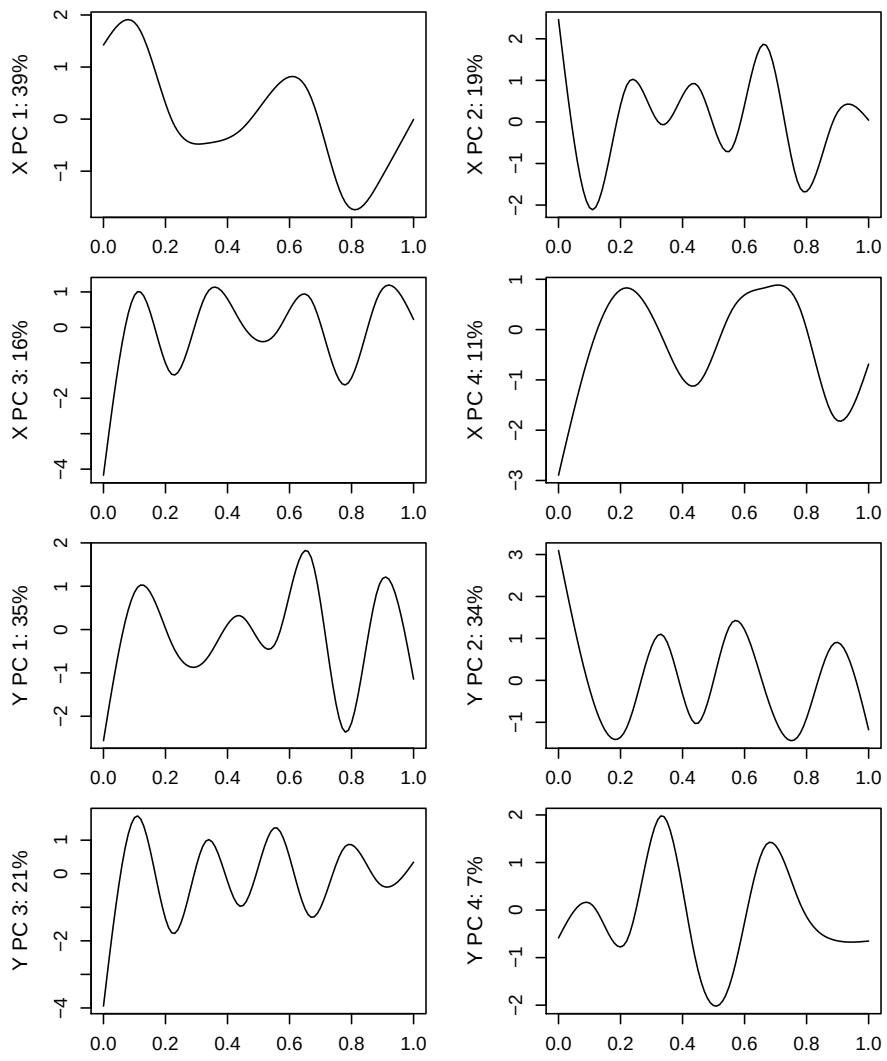

Figure 3.8: Case study 1. The first four eigenfunctions for predictor (upper four plots) and response (lower four plots).

and $s = 0.6$, the relationship is positive. The confidence band of β is wide in some area. One possible reason is for such functional predictor and response which have many fluctuation, the FPCA may not be an ideal basis function.

3.5.2 Case Study 2: Air Pollution

The this case study, we look into the air pollution data. The data set contains daily trajectories of oxides of nitrogen (NO_X) in the city of Sacramento, California, on 52 summer days in the year

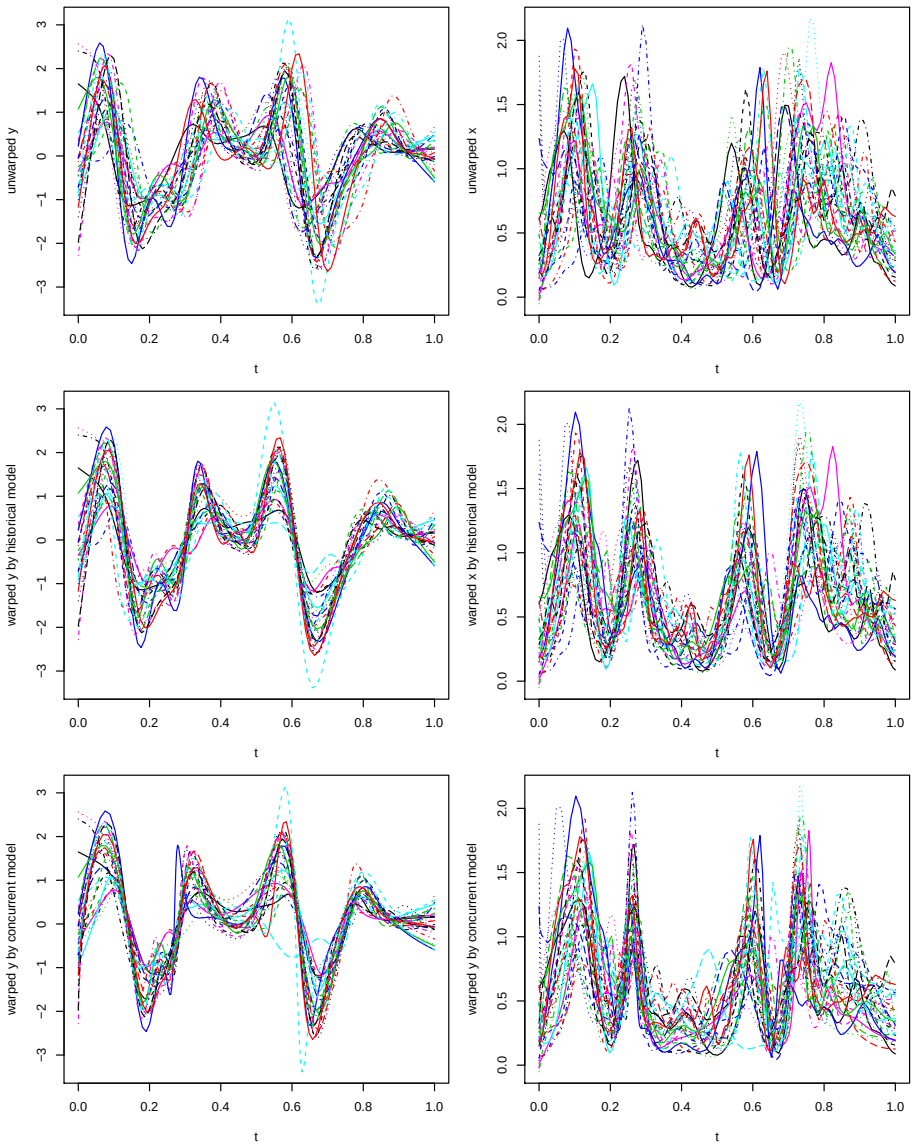

Figure 3.9: Case study 1. Warped observations by different models.

2005, and the corresponding trajectories of ozone concentration (O_3). O_3 is formed by a series of complex photochemical reactions between nitrogen oxides and volatile organic compounds in the presence of sunlight. The goal is to explore the relationship between NO_X and O_3 during a cycle of day.

Before fit the model, we first plotted the first four eigenfunctions of predictor and response, as shown in Figure 3.12. The first four eigenfunctions account for more than 90% if the total variations. When we fit the historical model, it is sufficient to set the number of PCA basis functions

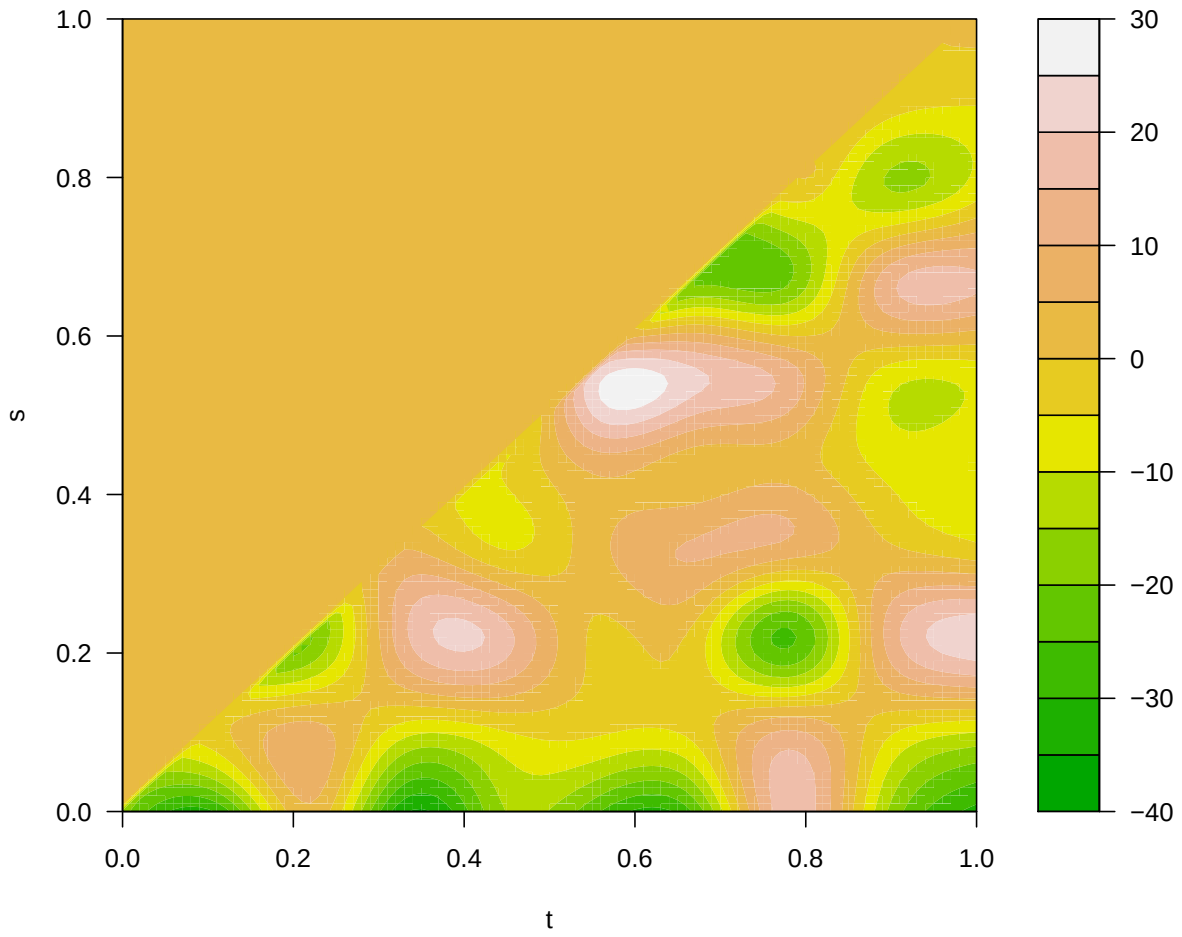

Figure 3.10: Case study 1. Contour plot of estimated $\beta(s,t)$.

$p = q = 4$. The plotted eigenfunctions also show where the most variation come from. For predictor, most variation comes from the time interval $0 - 0.4$, and for response its most variation comes from $0.4 - 0.8$.

We fit both historical model and concurrent model to the data, treating the square root of O_3 as response variable, and natural log transformation of NO_X as predictor. The concurrent model has much worse fit than historical model ($MSE_{concurrent} = 101$, $MSE_{historical} = 62$; $WAIC_{concurrent} = 70112, WAIC_{historical} = 45774$). Therefore, we choose historical model as our final model.

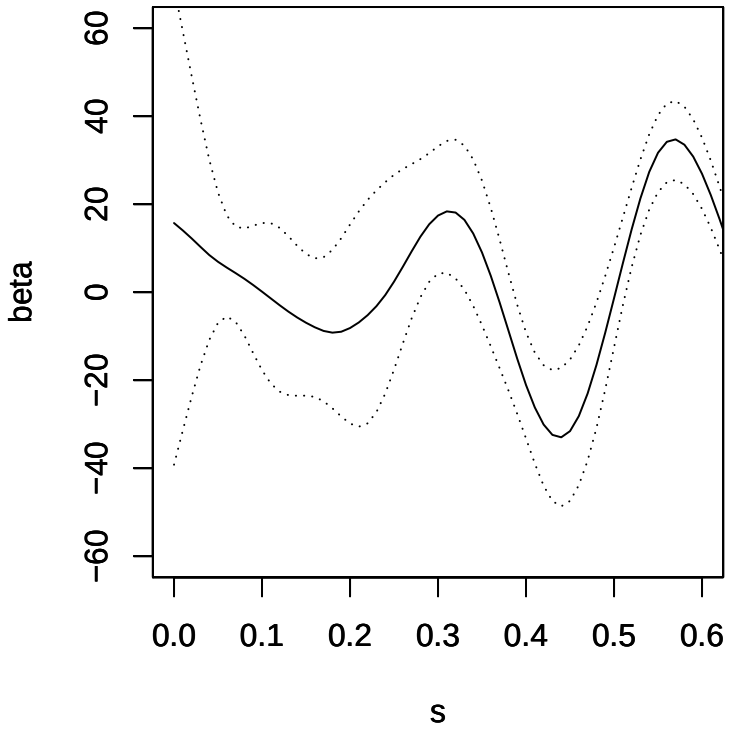

Figure 3.11: Case study 1. Estimated mean $\beta(s, t = 0.6)$ (solid line) and its 95% confidence band (dotted line).

Figure 3.13 shows the observed data and predicted y, indicating good model fit. Figure 3.14 shows the contour plot of regression function $\beta(s, t)$. For O_3 between the time 10 am to 7 pm, it has a strong positive relationship with NO_X during midnight to 10 am. In other words, the effect of NO_X has about 10 hour delay on O_3. Figure 3.15 shows the confidence band of regression function at the peak time $t = 0.6$ (2 pm). The peak of O_3 at 2pm is closely positively related to NO_X before 10 am. Such delayed association was noticed in [Agudelo-Castaneda et al., 2014], where daily trajectories of NO_X and O_3 were collected from 2006 to 2009 for an urban area at Brazil. In their data set, it was observed the concentration of O_3 increases after sunrise (7 am) and reached maximum around 3 pm. NO_X level peaked around 8 am. The delayed association between O_3 and NO_X is possibly due to the photochemical reaction by sunlight which transforms NO_X into O_3.

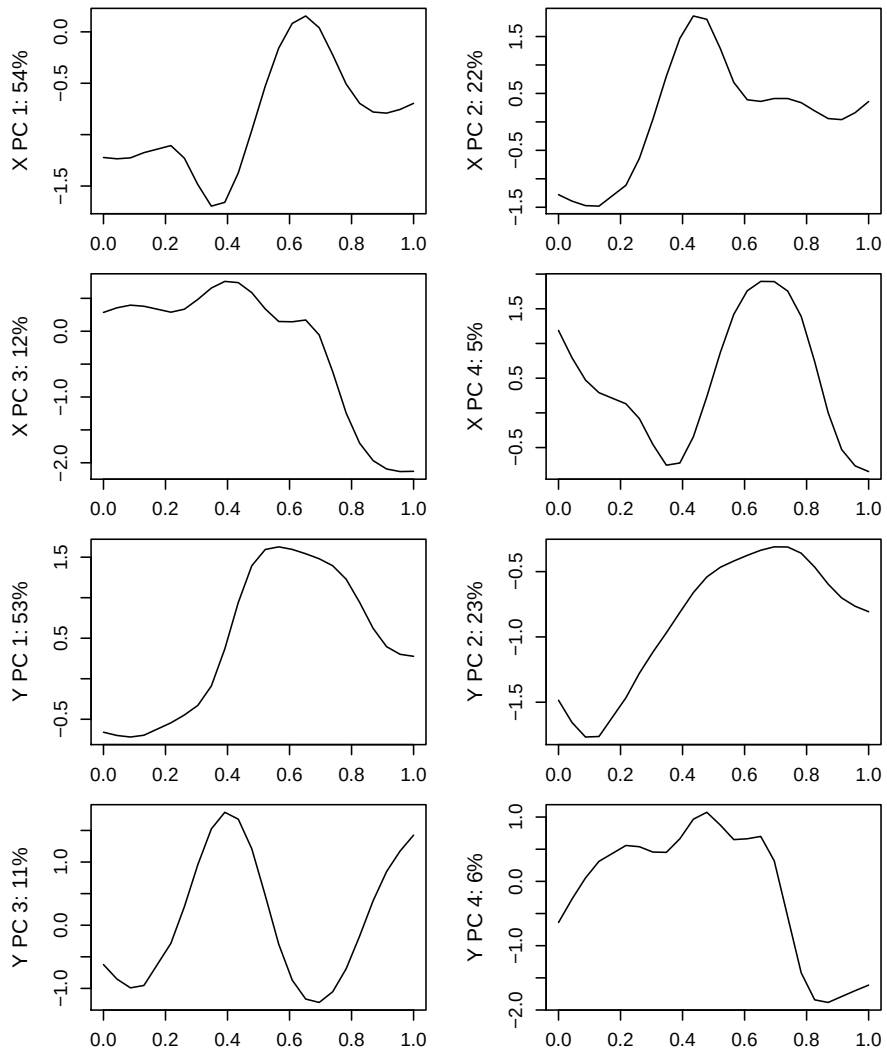

Figure 3.12: Case study 2. The first four eigenfunctions for predictor (upper four plots) and response (lower four plots).

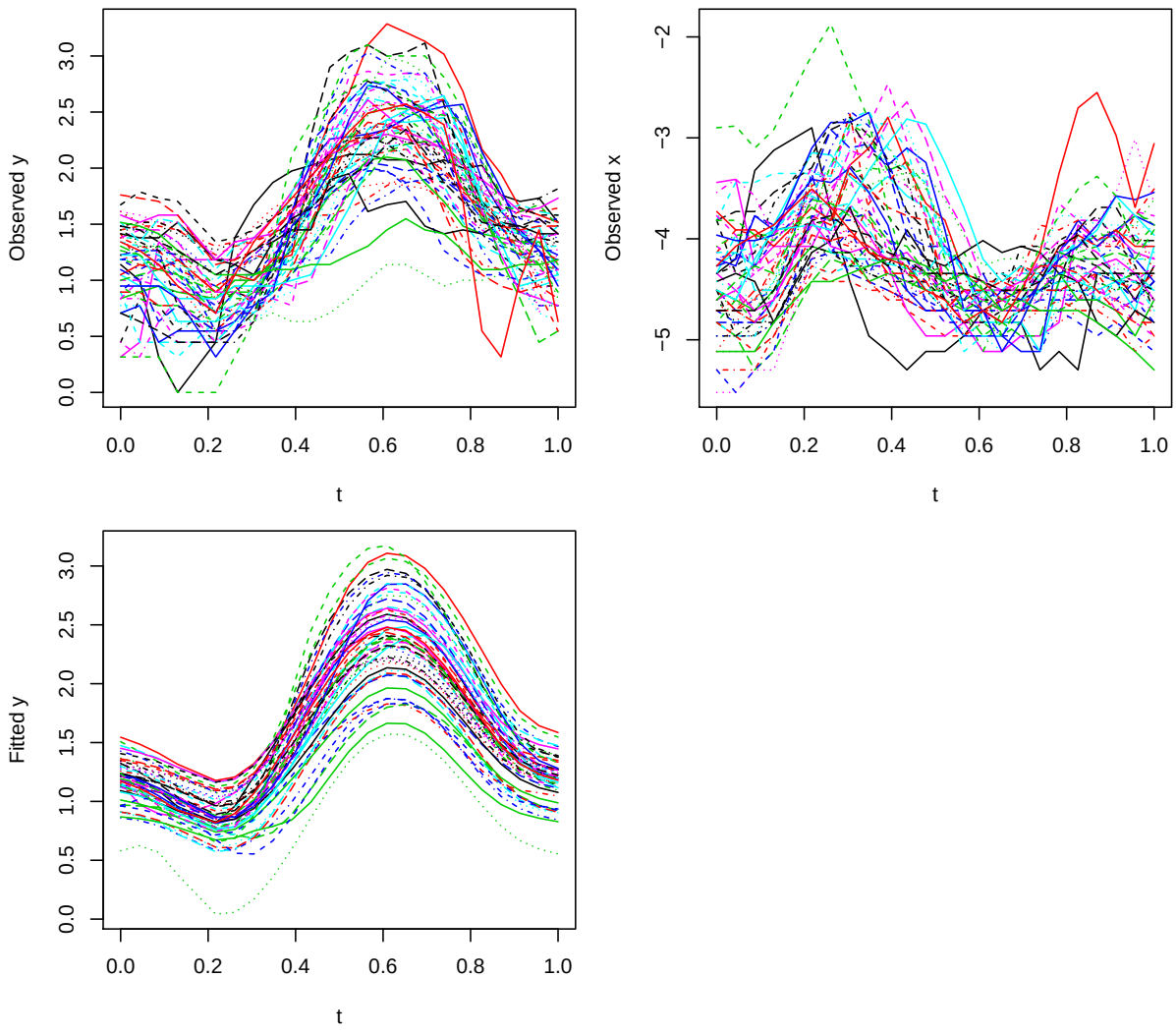

Figure 3.13: Case study 2. Observed and fitted data.

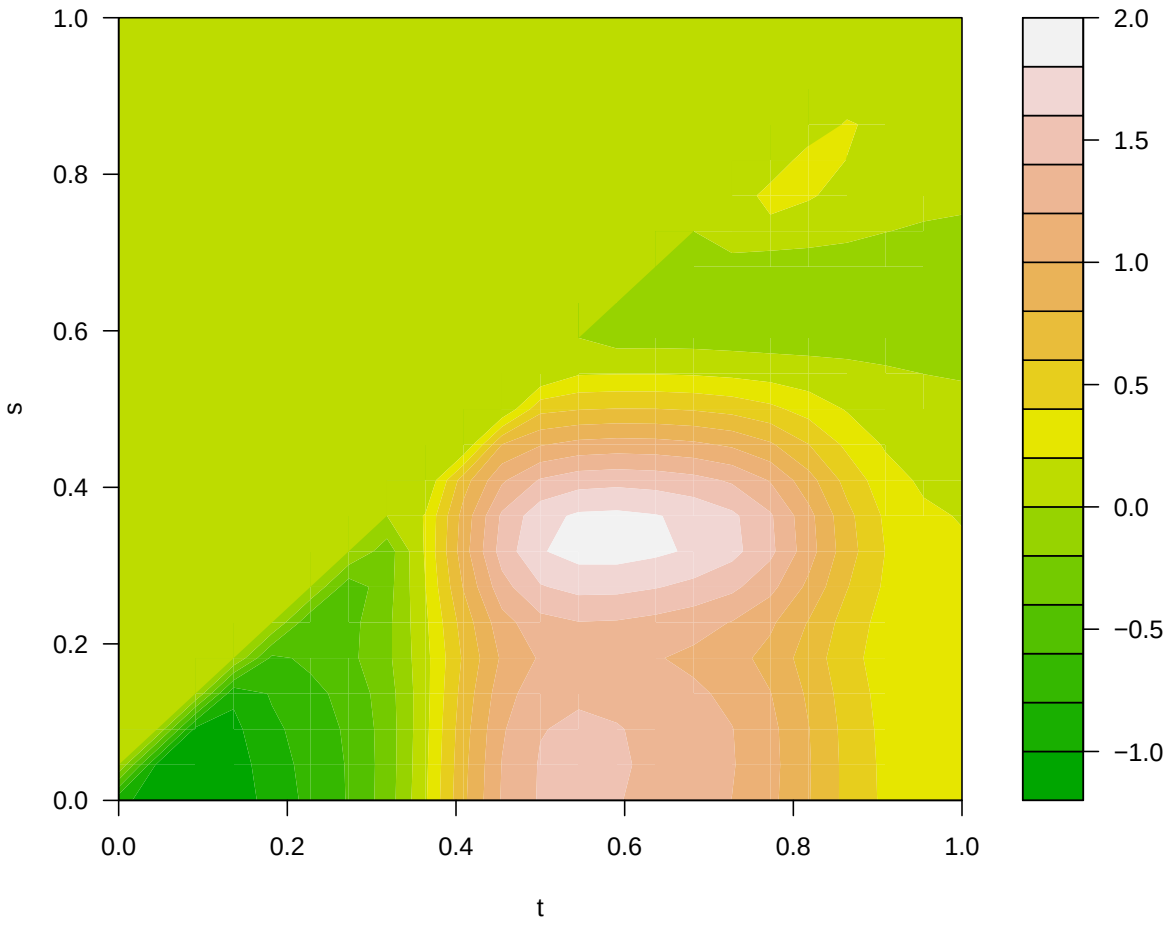

Figure 3.14: Case study 2. Contour plot of estimated $\beta(s,t)$. $0.4, 0.6, 0.8$ correspond to 10 am, 2 pm, and 7 pm.

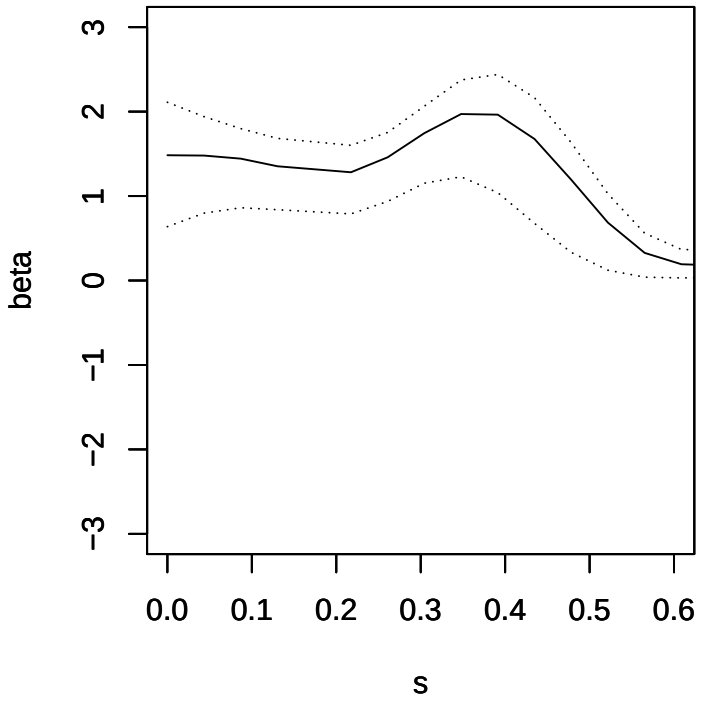

Figure 3.15: Case study 2. Estimated mean $\beta(s, t = 0.6)$ (solid line) and its 95% confidence band (dotted line).

3.6 Discussion

In this chapter we propose a new method to integrate curve registration and functional regression under a joint model. The key idea of this new method is to build the regression model upon warped response and predictor variables. Bayesian method is adopted for parameter estimation. We apply the method to two different function-on-function regression models: historical and concurrent model. The simulation results show the proposed method provides satisfactory goodness-of-fit for these two regression models.

Followings are the key components of the proposed method. The first one is the bivariate regression function $\beta(t, s)$, which represents the regression surface between the functional response and predictor. Estimation of $\beta(t, s)$ is obtained through decomposing it by functional PCA basis, which avoids identifiability issue. The second key component is the time transformation function $\mu_i(t)$. It is estimated through B-spline and Jupp transformation to maintain its monotonic feature. The third key component is that in each iteration when the new $\mu_i(t)$ is calculated, response and predictor variables need to be aligned according to current time transformation function. The alignment is done by spline interpolation.

In simulation study, we show that when time variation exists in predictor or response, functional regression without curve registration produces unreliable estimation. This emphasizes again the importance of handling time variation by appropriate model instead of ignoring it. We also compare historical model and concurrent model. They both work compatibly with the integrated time warping component. Historical model is able to relate remote areas of predictor and response, which can not be achieved by concurrent model. To visualize regression surface, contour plot is a useful too as shown in Figure 3.3. Another way to plot the result is by fixing t at a specific time point t_1, and the magnitude and sign of $\beta(s, t = t_1)$ can indicate which part of outcome is closely related to which part of predictor.

We apply the proposed method to air pollutant data to investigate potential relationship between O_3 and NO_X. The historical model has better model fit. By examining the regression surface, we find delayed association between NO_X and O_3. This finding shows the importance of historical model when the predictor and response are remotely related. We also apply this method to the

lip movement data. Evaluated by MSE and WAIC, historical model shows better model fit than concurrent model. The estimated regression surface is very complicated, which makes it hard to interpret.

The proposed method can be easily generalized to cases where multiple functional predictors are present, or multiple predictors are available in functional and scalar form. A more complicated scenario is that the predictor and response have different warping functions. [Gervini, 2014] proposed a joint model which includes time transformation parameters as predictor to solve such problem. It demonstrates that their method achieves good predictive power and allows unified statistical inference about phase and amplitude components.

3.7 Appendix: Full Conditionals

- Full conditional distribution for vectorized $\mathbf{B}$

  $(\mathbf{B}|\boldsymbol{Y}, \boldsymbol{\theta}_{-b_i}) \sim N(B_m; V_B)$, where
  $V_B = (\boldsymbol{S}'\boldsymbol{S}/\sigma_\epsilon^2 + \Sigma_B^{-1})^{-1}, \mathbf{B}_m = V_B\boldsymbol{S}'\boldsymbol{Y}/\sigma_\epsilon^2$

- Full conditional distribution for σ_ϵ^2

  $(\sigma_\epsilon^2|\boldsymbol{Y}, \boldsymbol{\theta}_{-\sigma_\epsilon^2}) \sim IG(a_\epsilon^*, b_\epsilon^*)$
  $a_\epsilon^* = a_\epsilon + mn/2, b_\epsilon^* = b_\epsilon + (\boldsymbol{Y} - S_B\mathbf{B})'(\boldsymbol{Y} - S_B\mathbf{B})$

- Full conditional distribution for σ_τ^2

  $(\sigma_\tau^2|\boldsymbol{y}, \boldsymbol{\theta}_{-\sigma_\tau^2}) \sim IG(a_\tau^*, b_\tau^*)$
  $a_\tau^* = a_\tau + nw/2, b_\tau^* = b_\tau + 1/2\sum(\boldsymbol{\tau}_i - \boldsymbol{\Upsilon})'\Omega_\tau(\boldsymbol{\tau}_i - \boldsymbol{\Upsilon})$, w is the length of $\boldsymbol{\tau}_i$.

CHAPTER 4

Summary and Future Development

4.1 Summary

In this work, we have presented two new methods which tackle two different aspects of functional data analysis. The first method is focusing on accelerating Bayesian curve registration, which is computationally demanding because of MCMC simulation. This is especially problematic if the number of evaluating points m is large. To solve such problem, the proposed method smoothes the curve y by k B-spline basis functions, and represents the coefficient of the B-splines by a registration process. This method, while seemingly overparametrized compared to the standard hierarchical curve registration model, leads to significant computational savings when $k << m$. Simulation study shows time cost of the new method is 36% of the time that standard BHCR costs. In the case study, we apply this new method to ICP data where curves are intensively sampled. The new method successfully aligns ICP curves and provides mean function for each patient. With these mean functions, we are able to draw summary statistics upon them for further analysis.

The second proposed method is Bayesian warped functional regression. The key idea of this method is to build functional regression model and perform curve registration simultaneously. To achieve this goal, in every iteration of MCMC sampling, the aligned response and predictor needs to be re-calculated through interpolation. Simulation study shows, when misalignment exists, warped functional regression has much better model fit than ordinary functional regression without curve registration. We implement this method on two types of regression: historical function-on-function regression and concurrent function-on-function regression. Simulation shows the model fit is satisfactory for both of them. The interpretation of regression result replies on fix regression function $\beta(s,t)$ at a specific time point t_1. The magnitude and sign of $\beta(s, t = t_1)$ indicates the how

predictor contributes to outcome. We then apply the Bayesian warped functional regression model on two case studies. For both of them the historical model has better model fir than concurrent model. Historical model is able to relate remote areas of predictor and response, which can not be achieved by concurrent model.

4.2 Future Development

An important assumption assumption in our proposed warped functional regression is that response and predictor variables share the same warping function. This may not be true in reality. When they have different warping functions, the estimation can be done in a joint model:

$$\begin{aligned} X_i(t) &= X_i^*(t) \circ \mu_i(t) + \epsilon_i(t) \\ Y_i(t) &= [B_0(t) + \int X_i^*(s)\beta(s,t)ds] \circ \eta_i(t) + E_i(t) \end{aligned} \tag{4.1}$$

where $\mu_i(t)$ and $\eta_i(t)$ are warping functions for x and y respectively. Moreover, the current warped functional regression can be extended from function-on-function regression to scalar-on-function: $Y_i = B_0 + \int X_i^*(t)\beta(t)dt + E_i$ and function-on-scalar: $Y_i(t) = [\sum_{j=1}^{p} X_{ij}\beta_j(t)] \circ \eta_i(t) + E_i(t)$.

Another possible extension to warped functional regression is to ease the computation cost of time warping by Variational Bayes method or the predictive model based method introduced in Chapter 2. Variational Bayes for curve registration problem has been introduced in [Earls and Hooker, 2015]. Since the parameter of time transformation function have non-conjugate priors (this is common for curve registration problem), the variational Bayes procedure is divided into two stages. The parameters with conjugate priors are estimated through standard variational Bayes formula, whereby the parameters with non-conjugate priors are estimated through maximizing a target function.

A third possible future development is to account for within-subject correlation when performing curve registration. Suppose we have a data set consisting of multiple subjects, and each subject have several trajectories observed for the same underlying unknown function. If our goal is to align curves from all subjects and estimate the overall mean function, we need to take within-subject correlation into account in terms of modeling the registration process. This is also a problem when

considering mixed functional regression model. We may add random functions to account for necessary covariance structures.

REFERENCES

[Agudelo-Castaneda et al., 2014] Agudelo-Castaneda, D. M., Teixeira, E. C., and Pereira, F. N. (2014). Time–series analysis of surface ozone and nitrogen oxides concentrations in an urban area at brazil. *Atmospheric Pollution Research*, 5(3):411–420.

[Albert and Chib, 1993] Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

[Banerjee et al., 2008] Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848.

[Bigelow and Dunson, 2007] Bigelow, J. L. and Dunson, D. B. (2007). Bayesian adaptive regression splines for hierarchical data. *Biometrics*, 63(3):724–732.

[Craven and Wahba, 1978] Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

[de Boor, 1978] de Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer.

[Denison et al., 1998] Denison, D. G. T., Mallick, B. K., and Smith, A. F. M. (1998). Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society, Series B*, 60(2):333–350.

[Di et al., 2009] Di, C.-Z., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. (2009). Multilevel functional principal component analysis. *Annals of Applied Statistics*, 3(1):458–488.

[DiMatteo et al., 2001] DiMatteo, I., Genovese, C. R., and Kass, R. E. (2001). Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071.

[Earls and Hooker, 2015] Earls, C. and Hooker, G. (2015). Adapted variational bayes for functional data registration, smoothing, and prediction. *arXiv preprint arXiv:1502.00552*.

[Eide, 2006] Eide, P. K. (2006). A new method for processing of continuous intracranial pressure signals. *Medical engineering & physics*, 28(6):579–587.

[Ferraty and Vieu, 2002] Ferraty, F. and Vieu, P. (2002). The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17(4):545–564.

[Fox and Dunson, 2012] Fox, E. and Dunson, D. B. (2012). Multiresolution gaussian processes. In *Advances in Neural Information Processing Systems*, pages 737–745.

[Friedman, 1991] Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–67.

[Gasser and Kneip, 1995] Gasser, T. and Kneip, A. (1995). Searching for structure in curve samples. *Journal of the American Statistical Association*, 90(432):1179–1188.

[Gasser et al., 1991] Gasser, T., Kneip, A., and Kohler, W. (1991). A flexible and fast method for automatic smoothing. *Journal of the American Statistical Association*, 86(415):643–652.

[George and Mcculloch, 1993] George, E. I. and Mcculloch, R. E. (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

[Gervini, 2014] Gervini, D. (2014). Warped functional regression. *Biometrika*, page asu054.

[Gervini, 2015] Gervini, D. (2015). Dynamic retrospective regression for functional data. *Technometrics*, 57(1):26–34.

[Gervini and Gasser, 2004] Gervini, D. and Gasser, T. (2004). Self-modelling warping functions. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 66:959–971.

[Goldsmith et al., 2012] Goldsmith, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2012). Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(3):453–469.

[Goldsmith et al., 2011] Goldsmith, J., Feder, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4):830–851.

[Green, 1995] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

[Greven et al., 2010] Greven, S., Crainiceanu, C., Caffo, B., and Reich, D. (2010). Longitudinal functional principal component analysis. *Electronic journal of statistics*, 4:1022.

[Hastie and Mallows, 1993] Hastie, T. and Mallows, C. (1993). [a statistical view of some chemometrics regression tools]: Discussion. *Technometrics*, 35(2):pp. 140–143.

[Hastie and Tibshirani, 1993] Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 757–796.

[Hastie and Tibshirani, 1990] Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Chapman and Hall.

[He et al., 2000] He, G., Müller, H., and Wang, J. (2000). Extending correlation and regression from multivariate to functional data. *Asymptotics in statistics and probability*, pages 197–210.

[Hu et al., 2009] Hu, X., Xu, P., Scalzo, F., Vespa, P., and Bergsneider, M. (2009). Morphological clustering and analysis of continuous intracranial pressure. *IEEE Transactions on Biomedical Engineering*, 56(3):696–705.

[Huang et al., 2004] Huang, J. Z., Wu, C. O., and Zhou, L. (2004). Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, pages 763–788.

[James, 2002] James, G. M. (2002). Generalized linear models with functional predictors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):411–432.

[James and Silverman, 2005] James, G. M. and Silverman, B. W. (2005). Functional adaptive model estimation. *Journal of the American Statistical Association*, 100(470):565–576.

[James et al., 2009] James, G. M., Wang, J., and Zhu, J. (2009). Functional linear regression that's interpretable. *The Annals of Statistics*, pages 2083–2108.

[Jupp, 1978] Jupp, D. L. (1978). Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 15(2):328–343.

[Kneip and Gasser, 1992] Kneip, A. and Gasser, T. (1992). Statistical tools to analyze data representing a sample of curves. *Annals of Statistics*, 20(3):1266–1305.

[Lang and Brezger, 2004] Lang, S. and Brezger, A. (2004). Bayesian p-splines. *Journal of Computational and Graphical Statistics*, 13(1):183–212.

[Lang et al., 2002] Lang, S., Fronk, E.-M., and Fahrmeir, L. (2002). Function estimation with locally adaptive dynamic models. *Computational Statistics*, 17(4):479–500.

[Luo and Wahba, 1997a] Luo, Z. and Wahba, G. (1997a). Hybrid adaptive splines. *Journal of the American Statistical Association*, 92(437):107–116.

[Luo and Wahba, 1997b] Luo, Z. and Wahba, G. (1997b). Hybrid adaptive splines. *Journal of the American Statistical Association*, 92(437):107–116.

[Malfait and Ramsay, 2003] Malfait, N. and Ramsay, J. O. (2003). The historical functional linear model. *Canadian Journal of Statistics*, 31(2):115–128.

[Morris and Carroll, 2006] Morris, J. S. and Carroll, R. J. (2006). Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199.

[Müller et al., 2008] Müller, H.-G., Chiou, J.-M., and Leng, X. (2008). Inferring gene expression dynamics via functional regression analysis. *BMC bioinformatics*, 9(1):60.

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

[Osborne et al., 1998] Osborne, M. R., Presnell, B., and Turlach, B. A. (1998). Knot selection for regression splines via the lasso. *Computing Science and Statistics*, pages 44–49.

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

[Rice and Silverman, 1991] Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society Series B-Methodological*, 53(1):233–243.

[Ruppert and Carroll, 2000] Ruppert, D. and Carroll, R. J. (2000). Theory & methods: Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223.

[Smith and Kohn, 1996] Smith, M. and Kohn, R. (1996). Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75:317–344.

[Sood et al., 2009] Sood, A., James, G. M., and Tellis, G. J. (2009). Functional regression: A new model for predicting market penetration of new products. *Marketing Science*, 28(1):36–51.

[Staniswalis and Lee, 1998] Staniswalis, J. G. and Lee, J. J. (1998). Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418.

[Telesca and Inoue, 2008] Telesca, D. and Inoue, L. Y. T. (2008). Bayesian hierarchical curve registration. *Journal of the American Statistical Association*, 103(481):328–339.

[Thompson and Rosen, 2008] Thompson, K. W. and Rosen, O. (2008). A bayesian model for sparse functional data. *Biometrics*, 64:54–63.

[Tuddenham and Snyder, 1954] Tuddenham, R. D. and Snyder, M. M. (1954). Physical growth of california boys and girls from birth to eighteen years. *Publications in child development. University of California, Berkeley*, 1(2):183.

[Yao et al., 2003] Yao, F., Müller, H.-G., Clifford, A. J., Dueker, S. R., Follett, J., Lin, Y., Buchholz, B. A., and Vogel, J. S. (2003). Shrinkage estimation for functional principal component scores with application to the population kinetics of plasma folate. *Biometrics*, 59(3):676–685.

[Yao et al., 2005] Yao, F., Müller, H.-G., Wang, J.-L., et al. (2005). Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903.

[Zellner, 1986] Zellner, A. (1986). *"On Assessing Prior Distributions and Bayesian Regression Analysis with g Prior Distributions" in Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*. Studies in Bayesian econometrics and statistics. North-Holland.

[Zhou and Shen, 2001] Zhou, S. G. and Shen, X. T. (2001). Spatially adaptive regression splines and accurate knot selection schemes. *Journal of the American Statistical Association*, 96(453):247–259.