1-21-2010

PENALIZED FUNCTIONAL REGRESSION

Jeff Goldsmith
Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics, jgoldsmi@jhsph.edu

Jennifer Feder
Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics

Ciprian M. Crainiceanu
Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics

Brian Caffo
Johns Hopkins Bloomberg School of Public Health, Department of Biostatistics

Daniel Reich
Translational Neuroradiolgy Unit, Neuroimmunology Branch, National Institute of Neurological Disorders and Stroke, Johns Hopkins University, Department of Neurology

Penalized Functional Regression

Jeff Goldsmith, Jennifer Feder, Ciprian M. Crainiceanu,
Brian Caffo and Daniel Reich

20 January 2010

Abstract

We develop fast fitting methods for generalized functional linear models. An undersmooth of the functional predictor is obtained by projecting on a large number of smooth eigenvectors and the coefficient function is estimated using penalized spline regression. Our method can be applied to many functional data designs including functions measured with and without error, sparsely or densely sampled. The methods also extend to the case of multiple functional predictors or functional predictors with a natural multilevel structure. Our approach can be implemented using standard mixed effects software and is computationally fast. Our methodology is motivated by a diffusion tensor imaging (DTI) study. The aim of this study is to analyze differences between various cerebral white matter tract property measurements of multiple sclerosis (MS) patients and controls. While the statistical developments proposed here were motivated by the DTI study, the methodology is designed and presented in generality and is applicable to many other areas of scientific research. An online appendix provides R implementations of all simulations.

1 Introduction

Unarguably, advancements in technology and computation have led to a rapidly increasing number of applications where measurements are functions or images. These developments have been accompanied and, in some cases, anticipated by intense methodological development in regression models where some covariates are functions Cardot et al. (2003); Cardot and Sarda (2005); Crainiceanu et al. (2008); Ferraty and Vieu (2006); James (2002); Muller and Stadtmuller (2005); Reiss and Ogden (2007); Ramsay and Silverman (2005). In this paper we develop a novel inferential approach to functional regression. Our goals are to:

1) simplify the methodology by reducing the number of tuning parameters; 2) increase the spectrum of models and applications where functional regression can be applied automatically; and 3) produce software that is fast and easy to generalize to more complex data and models. These goals are achieved by smoothing the covariance operators, using a large number of eigenvectors to capture the variability of the functional predictors, and modeling the functional regression parameters as penalized splines. The level of smoothing is estimated using Restricted Maximum Likelihood (REML) or cross-validation in an associated mixed effect model. Methods are implemented using standard mixed effects software.

An important advantage of our penalized functional regression (PFR) approach is that it is designed for a wider class of problems than other published methods. In particular, it applies to cases when functions are measured with or without error, at equal or unequal intervals, at a dense or sparse set of points. Moreover, methods apply to outcomes distributed in the exponential family class of models and to multiple functional regressors observed at one or multiple levels. A second advantage is that our methodology allows the automatic construction of confidence intervals using the mixed effects inferential machinery. A third advantage is that our software is very fast and scales to very large data sets.

Briefly, functional regression seeks to quantify the relationship between a scalar outcome and a functional regressor. To illustrate the main ideas, we start with the simple example when univariate functional data are measured at a single level. More specifically, assume that for each subject, $i = 1, \ldots, I$, we observe data $[Y_i, X_i(t), \mathbf{Z}_i]$, where Y_i is a scalar outcome, $X_i(t) \in \mathcal{L}^2[0,1]$ are random functions, and $\mathbf{Z}_i$ is a vector of nonfunctional covariates. We call $X_i(t)$ "univariate" functional data because in this example we only consider one functional regressor. The case of multivariate functional regressors is considered in Section 3.1. Moreover, we call $X_i(t)$ a "single level" sample of random functions, because only one function, $X_i(t)$, is sampled per subject. A multilevel or clustered case is considered in Section 3.2. The generalized functional linear model relating Y_i to the covariates $X_i(t), \mathbf{Z}_i$ is given by Cardot and Sarda (2005); McCullagh and Nelder (1989); Muller and Stadtmuller (2005)

$$\begin{aligned} Y_i &\sim \text{EF}(\mu_i, \eta) \\ g(\mu_i) &= \alpha + \int_0^1 X_i(s)\beta(s)ds + \mathbf{Z}_i\boldsymbol{\gamma} \, . \end{aligned} \tag{1}$$

Here $EF(\mu_i, \eta)$ denotes an exponential family distribution with mean μ_i and dispersion parameter η, $g(\cdot)$ is a link function, and $\beta(t) \in \mathcal{L}^2[0,1]$. The functional regression model is a powerful and practical inferential tool, in spite of the fact that observations $X_i(t)$ are never truly functional. Rather, we observe $\{X_i(t_{ij}) : t_{ij} \in [0,1]\}$, with $j = 1, \ldots, J_i$. Further,

the regressor functions are often measured with error; that is, one often measures a proxy functional covariate, $W_i(t) = X_i(t)+\epsilon_i(t)$, where $\epsilon_i(t)$ is a mean-zero white noise process with variance σ^2_ϵ. Thus, for subject i, data typically are of the form $[Y_i, \{W_i(t_{ij}) : t_{ij} \in [0,1]\}, \mathbf{Z}_i]$, $i = 1, \ldots, I$, $j = 1, \ldots, J_i$. In practice, functional data will have various sampling schemes. For example, t_{ij}, $j = 1, \ldots, J_i$ could be equally or unequally spaced for each subject, sparse at the subject level and dense at the population level, or dense at the subject and population level. The functions $X_i(t)$ can be measured with no, moderate or large measurement error.

Of interest are all the parameters of model (1) including the function $\beta(\cdot)$, which characterizes the relationship between the transformed mean of Y and the covariate of interest $X(\cdot)$. Before diving further into technical details it is worth explaining the interpretation of model (1). Indeed, we have found difficulty in interpretation to be the major hurdle for the adoption of such models by our collaborators. The core issue is that explaining the integral $\int X_i(s)\beta(s)ds$ is not straightforward. Consider a fine grid of points $s_1, \ldots, s_G$ in $[0,1]$ and approximate the integral by a Riemann sum

$$\int_0^1 X_i(s)\beta(s)ds \approx \frac{1}{G}\sum_{g=1}^{G} X_i(s_g)\beta(s_g) = \sum_{g=1}^{G} X_i(s_g)\{\beta(s_g)/G\}.$$

Thus, the $\beta(\cdot)$ function is, basically, re-weighting all subject level functions, $X_i(\cdot)$, using the weights $\beta(s_g)/G$. Each $\beta(\cdot)$ function, or equivalently, each weighting scheme, will tend to emphasize certain parts of the functional regressor more than others. An extreme example is when $\beta(\cdot) = \beta$, that is all $X_i(\cdot)$ observations receive the same weight. In this case $\int_0^1 X_i(s)\beta(s)ds = \beta\int_0^1 X_i(s)ds$ and model (1) becomes a standard regression model which contains the average functional covariate as a regressor. A less extreme example is when $\beta(t) = \beta$ if $t \leq 0.5$ and 0 otherwise. In this case $\int_0^1 X_i(s)\beta(s)ds = \beta\int_0^{0.5} X_i(s)ds$ and model (1) becomes a standard regression model which contains the average functional covariate over the interval $[0, 0.5]$ as a regressor. This considers all functional observations in the interval $[0, 0.5]$ equally important and all those in $[0.5, 1]$ equally useless. Intuition could further be enhanced by contemplating the interpretation of other simple step functions. In practice it makes sense to consider a smoother transition in the weighting scheme, that is a smooth $\beta(\cdot)$ function. We found the following interpretation of $\int_0^1 X_i(s)\beta(s)ds$ useful, albeit imperfect:

> *The subject-specific random variable that is most predictive of the outcome obtained by re-weighting each subject-specific curve by the same, population level, weights, $\beta(\cdot)$. Weights close to zero de-emphasize subject-level areas that are not predictive of the outcome, while large relative weights emphasize areas of the*

curve that are most predictive of the outcome. The collection of random variables $\int_0^1 X_i(s)\beta(s)ds$ *can be interpreted as a population of scores or indexes; comparing the outcomes for subjects with scores in the upper and lower quantiles of this distribution could be used to illuminate the relationship between the outcome and the functional variates.*

Our proposed approach to estimating the coefficient function $\beta(t)$ has two steps (the following uses notation in Ramsay and Silverman (2005), Ch. 15). First, we estimate the random functions using a finite series expansion $X_i(t) = \sum_{j=1}^{K_x} c_{ij}\psi_j(t)$, where $\boldsymbol{\psi} = \{\psi_1(t), ..., \psi_{K_x}(t)\}$ is the collection of the first K_x eigenfunctions of the smoothed covariance matrix $K^X(s,t) = \text{cov}[X_i(s), X_i(t)]$. Second, we use a truncated power series spline basis $\boldsymbol{\phi}(t) = \{\phi_1(t), ..., \phi_{K_b}(t)\}$ for $\beta(t)$, so that $\beta(t) = \boldsymbol{\phi}(t)\mathbf{b}$. The truncated power series representation of $\beta(t)$ imposes differentiability and allows simple control of smoothness. The tuning parameters, K_x and K_b, are considered to be very important in practice and their choice has been extensively debated in the functional and smoothing literature, respectively. In the smoothing literature, the choice of the number of knots has been shown Li and Ruppert (2008); Ruppert (2002) to be unimportant as long as it is large enough to capture the maximum complexity of the regression function. In penalized spline regression it is the smoothing parameter that takes care of reducing the variability of the functional estimate and avoids the heavy computational costs associated with choosing the number and positions of knots. We emulate this principle in the current functional setting and choose K_b large; typically we set $K_b = 35$. The choice of the number of eigenfunctions K_x is subject to the identifiability constraint $K_x \geq K_b$, so we choose and fix $K_x = \min\{35, M\}$, where M is the dimension of $K^X(\cdot,\cdot)$. That is, by choosing a large number we avoid the choice of a "good" number of principal components (PCs). Once the bases for $X_i(t)$ and $\beta(t)$ and the parameters K_x and K_b have been selected, model (1) may be expressed as a generalized linear mixed effects model (GLMM); thus the GLMM inferential machinery can be applied.

Our methods are most closely related to the functional regression framework developed in Cardot et al. (2003); Cardot and Sarda (2005), who proposed a penalized spline to estimate the functional parameter. We incorporate this idea but expand its scope to functions $X_i(t)$ that are measured with error or are sparsely sampled; this is achieved by using a PC basis to expand $X_i(t)$. The same idea can be extended seamlessly to functional regression when the exposure proxy has a multilevel structure Crainiceanu et al. (2008); Di et al. (2008). By making and exploiting the connection to mixed effects models we also provide a natural framework for necessary generalizations. Our methods are much faster because we take

advantage of the link to mixed effects models and existing well tested software. The modularity of our approach leads to straightforward extensions and seamless integration with other popular regression frameworks.

It is important we distinguish our use of the PC decomposition from the widely used functional principal components regression (FPCR) techniques Cardot et al. (1999); Reiss and Ogden (2007). Stated shortly, FPCR regresses the vector of scalar outcomes Y on the design matrix $\mathbf{X}\mathbf{V}_{\mathrm{A}}$, where $\mathbf{X}$ has i^{th} row $[X_i(t_1), ..., X_i(t_T)]$, $\mathbf{V}_{\mathrm{A}}$ is the truncated at A version of the matrix $\mathbf{V}$ in $\mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$, the singular value decomposition of $\mathbf{X}$. That is, FPCR regresses Y on the first A PC loadings of the functional regressors. In contrast, we use the PC decomposition only to provide estimates of the functional covariates using a small number of eigenfunctions. Indeed, using decompositions of the functional covariates in terms of other bases in the PFR method is straightforward.

The paper is organized as follows. Section 2, provides the details of our approach to functional regression. Section 3 describes the seamless generalization to multiple and clustered functions, and Section 4 describes the generalization to sparse functional data. Section 5 provides a detailed simulation to compare these methods. We apply our method to the DTI data in Section 6, and conclude with a discussion in Section 7. To ensure reproducibility of our results we post code for all simulations at http://biostat.jhsph.edu/∼jgoldsmi/Downloads /Web_Appendix_PFR.zip.

2 Approach

In this section we describe the PFR method for estimating the functional exposure effect $\beta(t)$. We focus first on estimating the subject specific functional effect, $X_i(t)$, and then we describe the estimators of $\beta(t)$ and its variability, respectively.

2.1 Estimation of $X_i(t)$

The first step in our analysis is to estimate, or predict, $X_i(t)$ in model 1 using an expansion into the PC basis obtained from its covariance operator, $K^X(\cdot, \cdot)$. As mentioned in Section 1, the problem of choosing the number of components is avoided by choosing a large number of PCs. This re-focuses the problem on estimating $K^X(\cdot, \cdot)$, which is a much simpler problem.

Assume that instead of observing $X_i(t)$ one measures a proxy $W_i(t) = X_i(t) + \epsilon_i(t)$, where $\epsilon_i(t)$ is a mean-zero white noise process with variance σ^2_ϵ. The covariance operator for the observed data is $K^W(s,t) = K^X(s,t) + \sigma^2_\epsilon \delta_{ts}$, where $K^W(s,t) = \mathrm{Cov}\{W_i(s), W_i(t)\}$

is the covariance operator on the observed functions, $K^X(s,t) = \text{Cov}\{X_i(s), X_i(t)\}$, and $\delta_{ts} = 1$ if $t = s$ and is 0 otherwise. This suggests the following strategy for estimating $K^X(s,t)$. First, construct a method of moments estimator $\hat{K}^W(s,t)$ of $K^W(s,t)$ from the observed data. Second, smooth $\hat{K}^W(s,t)$ for $s \neq t$, as suggested by Staniswalis and Lee (1998); Yao et al. (2003). The only serious problem we encountered in practice occurred when the functions $X_i(t)$ are unevenly or sparsely sampled. Consider the case when each pair of sampling locations, (t_{ik}, t_{il}), is unique. In this situation $K^W(t_{ik}, t_{il})$ is estimated by $\{W_i(t_{ik}) - W_i(t_{il})\}^2/2$; the number of pairs (t_{ik}, t_{il}) can quickly explode making bivariate smoothing of the estimated covariance matrix difficult. To avoid this problem we use the ideas suggested in Di et al. (2008) to estimate $K^W(s,t)$

1. Use a very small bandwidth smoother to obtain an undersmooth estimate of the covariance operator.

2. Use a fast automatic nonparametric smoother of the undersmooth surface obtained at the previous step.

Let $\sum_{k=1}^{\infty} \lambda_k \psi_k(s)\psi_k(t)$ be the spectral decomposition of $\hat{K}^X(s,t)$, where $\lambda_1 \geq \lambda_2 \geq \ldots$ are the non-increasing eigenvalues and $\boldsymbol{\psi}(\cdot) = \{\psi_k(\cdot) : k \in \mathbb{Z}^+\}$ are the corresponding orthonormal eigenfunctions. An approximation for $X_i(t)$, based on a truncated Karhunen-Loeve decomposition, is given by $X_i(t) = \sum_{k=1}^{K_x} c_{ik}\psi_k(t)$, where K_x is the truncation lag and $c_{ik} = \int_0^1 X_i(t)\psi_k(t)dt$. Unbiased estimators of c_{ik} are easy to obtain as the Riemann sum approximation to the integral $\int_0^1 W_i(t)\psi_j(t)dt$; for example, $\widehat{c}_{ik} = \sum_{j=1}^{J_i} W_i(t_{ij})\psi_k(t_{ij})$ was proposed by Muller and Stadtmuller (2005). This method works well when data are densely sampled and each subject-specific function is sampled at many points J_i. When this is not the case a better alternative is to obtain best linear unbiased predictors (BLUP) or posterior modes in the mixed effects model Crainiceanu et al. (2008); Di et al. (2008)

$$\begin{array}{rcl} W_i(t_{ij}) & = & \sum_{k=1}^{K_x} c_{ik}\psi_k(t_{ij}) + \epsilon_{ij} \\ c_{ik} & \sim & \text{N}(0, \sigma_c^2),\ \epsilon_{ij} \sim \text{N}(0, \sigma_\epsilon^2), \end{array} \tag{2}$$

where c_{ik} and ϵ_{ij} are mutually independent for every i, j, k. The subject-specific processes $X_i(t)$ are then predicted at *any* t by plugging-in the predictors of c_{ij} in the equality $X_i(t) = \sum_{k=1}^{K_x} c_{ik}\psi_k(t)$. A potential criticism of this method is that c_{ij} could be predicted with sizeable error which can lead to sizeable variability in the prediction of $X_i(t)$. In some situations this can lead to bias in the functional regression, as discussed in Crainiceanu et al. (2008). When this problem is a real concern a solution is to jointly model the outcome model and model (2).

This approach can be addressed using a fully Bayesian analysis Crainiceanu and Goldsmith (2009) and is not the focus of this paper. Instead, we focus on the two stage approach, which is the current state-of-the-art in functional regression.

We emphasize that the PC decomposition in the first step of our analysis is used to estimate the $X_i(t)$ when they are measured with error or sparsely sampled, rather than to address the ill-posed nature of the functional regression, as in FPCR. Thus, we focus the problem on estimating $\beta(t)$ using a method that does not depend on the particular choice of number of principal components, a non-trivial distinction.

2.2 Estimation of $\beta(t)$

The second step in our method is modeling $\beta(t)$ and we borrow ideas from the penalized spline literature O'Sullivan (1986); Ruppert et al. (2003); Wood (2006). Let $\boldsymbol{\phi}(t) = \{\phi_1(t), \phi_2(t), ..., \phi_{K_b}(t)\}$ be a spline basis, so that $\beta(t) = \sum_{k=1}^{K_b} b_k \phi_k(t) = \boldsymbol{\phi}(t)\mathbf{b}$, where $\mathbf{b} = \{b_1, \ldots, b_{K_b}\}^T$. Thus, the integral in model (1) becomes

$$\int_0^1 X_i(s)\beta(s)ds = \int_0^1 \mathbf{c}_i' \boldsymbol{\psi}^T(s)\boldsymbol{\phi}(s)\mathbf{b}\, ds = \mathbf{c}_i' \mathbf{J}_{\psi\phi}\mathbf{b},$$

where $\mathbf{c}_i' = (c_{i1}, \ldots, c_{iK_x})^T$, $\mathbf{J}_{\psi\phi}$ is a $K_x \times K_b$ dimensional matrix with the (k,l)th entry equal to $\int_0^1 \psi_k(s)\phi_l(s)ds$ Ramsay and Silverman (2005).

It would be mathematically simpler to expand $\beta(\cdot)$ in the principal component basis used for expanding the functional data, $\psi_1(\cdot), \ldots, \psi_{K_x}(\cdot)$. In spite of its apparent appeal, this approach is not satisfactory in many applications. The main technical reasons are that: 1) the principal component basis is typically not a parsimonious basis for the smooth parameter function; and 2) the smoothing of the $\beta(\cdot)$ function is implicitly controlled by K_x, the smoothing parameter for the functional process, $X_i(t)$. Thus, we use the a spline basis expansion $\beta(t) = \boldsymbol{\phi}(t)\mathbf{b}$ and induce smoothing by assuming that $\mathbf{b} \sim \mathrm{N}(0, \mathbf{D})$, where $\mathbf{D}$ is a penalty matrix corresponding to the particular spline basis $\boldsymbol{\phi}(t)$. The expression $\mathbf{b} \sim \mathrm{N}(0, \mathbf{D})$ contains a slight abuse of notation for the case when some of the $\mathbf{b}$ parameters are not penalized. For example, the assumption that b_1 is not penalized can be conceptually written as $b_1 \sim N(0, \infty)$.

Denote by $\mathbf{C}$ the $I \times K_x$ dimensional matrix with the ith row equal to $\mathbf{c}_i'$ and by $\mathbf{Z}$ the $I \times p$ dimensional matrix with the ith row equal to $\mathbf{Z}_i$. The outcome model (1) can be

reformulated in matrix format as:

$$\begin{aligned} \mathbf{Y} \mid \mathbf{X}(\mathbf{t}) &\sim EF(\boldsymbol{\mu}, \boldsymbol{\gamma}) \\ g(\boldsymbol{\mu}) &= [1\ \mathbf{CJ}_{\psi\phi}\ \mathbf{Z}][\alpha\ \mathbf{b}\ \boldsymbol{\gamma}]^T \\ \mathbf{b} &\sim \mathrm{N}(0, \mathbf{D}), \end{aligned}$$

which is a mixed effect model with K_b random effects, $\mathbf{b}$. This model can be fit robustly using standard mixed effects software.

Model (3) depends on the choice of basis for $\beta(t)$, K_b and K_z. While any penalized spline approach could be used, in this paper we use a truncated power series basis with K_b knots for $\beta(t)$. Following Ruppert (2002) we select $K_b = 35$, which is large enough to prevent undersmoothing in many applications, and select $K_x \geq K_b$. However, as noted, the specific value of K_b is unimportant as long as it is large enough to capture the maximum variability in $\beta(t)$; in some future applications it might be necessary to increase K_b, K_x. The position of the knots is typically unimportant and we place them at the quantiles of the distribution of t_{ij}.

It is worth noting that the complexity of fitting model (3) is the same as the complexity of fitting a penalized spline model with K_b random coefficients, a well researched problem with well-developed accompanying software Wood (2006).

2.3 Confidence intervals for $\beta(t)$

Because model (3) is a mixed effects model the typical inferential machinery for mixed effects models can be used to obtain variance-covariance estimates of the model parameters. Variance estimators, pointwise and joint confidence intervals can be obtained following standard methods and software Ruppert et al. (2003); Wood (2006).

For illustration, consider the case when $Y_i = \alpha + \int_0^T X_i(t)\beta(t)\,dt + \epsilon_i$ with $\epsilon_i \sim \mathrm{N}[0, \sigma_\epsilon^2]$. Take as the basis for $\beta(t)$ the functions $t, t^2, (t-\kappa_1)_+^2, ..., (t-\kappa_{K_b})_+^2$. Let $\boldsymbol{\beta} = [\alpha\ \mathbf{b}^T]^T$; it is easy to show that $\hat{\boldsymbol{\beta}} = (\mathbf{W}^{\mathrm{T}}\mathbf{W} + \lambda^2\mathbf{D})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{Y}$ where λ is the smoothing parameter, $\mathbf{W} = [1\ \mathbf{CJ}_{\psi\phi}]$ and

$$\mathbf{D} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{K\times 3} & \mathbf{1}_{K\times K} \end{bmatrix}.$$

The smoothing parameter $\lambda^2 = \sigma_\epsilon^2/\sigma_{\mathbf{b}}^2$ can be estimated via REML in the corresponding mixed effects model (3). Recall that $\beta(t) = \boldsymbol{\phi}(t)\mathbf{b}^T$. Let

$$\begin{aligned} \mathrm{Var}[\hat{\boldsymbol{\beta}}] &= \sigma_\epsilon^2(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \lambda^2\mathbf{D})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{W}(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \lambda^2\mathbf{D})^{-1} \\ &= \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\mathbf{b}\alpha} \\ \Sigma_{\mathbf{b}\alpha}^T & \Sigma_{\mathbf{bb}} \end{bmatrix}. \end{aligned}$$

Then, at any t_0 $\text{Var}[\hat{\beta}(t_0)] = \text{Var}[\boldsymbol{\phi}(t_0)\hat{\mathbf{b}}^T] = \boldsymbol{\phi}(t_0)\Sigma_{\mathbf{bb}}\boldsymbol{\phi}(t_0)^T$; we estimate $\widehat{\text{sd}}\{\hat{\beta}(t)\} = \sqrt{\boldsymbol{\phi}(t_0)\widehat{\Sigma}_{\mathbf{bb}}\boldsymbol{\phi}(t_0)^T}$, where $\widehat{\Sigma}_{\mathbf{bb}}$ is the $(K_b+2)\times(K_b+2)$ dimensional matrix obtained by plugging in the REML estimate for λ into the formula for $\text{Var}[\hat{\boldsymbol{\beta}}]$.

An approximate 95% confidence interval $\text{E}[\hat{\beta}(t_0)]$ can be constructed as $\hat{\beta}(t_0)\pm 1.96\,\widehat{\text{sd}}\{\hat{\beta}(t)\}$. If $\text{E}[\hat{\beta}(t_0)] \approx \beta(t_0)$, that is if bias is negligible, the above interval will approximate well the 95% CI for $\beta(t_0)$. For bias corrected confidence intervals see, for example, Ruppert et al. (2003).

3 Multivariate and multilevel extensions

3.1 Multivariate extensions

In this section, we extend our model to the case of multiple functional regressors. Suppose our observed data for subject i is of the form $[Y_i, \mathbf{Z}_i, \{W_{il}(t), t\in[0,1]\}]$, where Y_i is continuous or discrete, $\mathbf{Z}_i$ is a vector of covariates and $W_{il}(t)$, $1\le l\le L$, are the observed proxies for the true functional regressors $X_{ij}(t)$. We emphasize that the $X_{il}(t)$ are distinct functional regressors; a notationally similar - but conceptually different - setting is considered in the next section. A multivariate extension of our regression model (1) is given by

$$\begin{aligned} Y_i &\sim \text{EF}(\mu_i,\eta) \\ g(\mu_i) &= \alpha + \int_0^1 X_{i1}(s)\beta(s)ds + ... + \int_0^1 X_{iL}(s)\beta(s)ds + \mathbf{Z}_i\boldsymbol{\gamma}\,. \end{aligned} \tag{3}$$

The approach given in Section 2 extends naturally to the multivariate functional regression setting. For each functional regressor, we estimate $K_l^X(s,t) = \text{Cov}\{X_{il}(s), X_{il}(t)\}$ by smoothing the off-diagonal elements of the observed covariance operator $K_l^W(s,t) = \text{Cov}\{W_{il}(s), W_{il}(t)\}$. Let $\sum_{k=1}^{\infty}\lambda_{kl}\psi_{kl}(s)\psi_{kl}(t)$ be the spectral decomposition of $\hat{K}_l^X(s,t)$, where $\lambda_{1l}\ge\lambda_{2l}\ge ...$ are the non-increasing eigenvalues and $\boldsymbol{\psi}_l(\cdot) = \{\psi_{kl}(\cdot) : k\in\mathbb{Z}^+\}$ are the corresponding eigenfunctions. We approximate $X_{il}(t)$ using a truncated Karhunen-Loéve decomposition so that $X_{il}(t) = \sum_{k=1}^{K_x} c_{ikl}\psi_{kl}(t)$, where K_x is the truncation lag and $c_{ikl} = \int_0^1 X_{il}(t)\psi_{kl}(t)dt$; unbiased estimators of c_{ikl} are given by $\hat{c}_{ikl} = \sum_{j=1}^{J} W_{il}(t)\psi_{kl}(t)$.

As in Section 2, express each coefficient function in model (3) in terms of a spline basis $\boldsymbol{\phi}(t) = \{\phi_1(t), \phi_2(t), ..., \phi_{K_b}(t)\}$, so that $\beta_l(t) = \sum_{k=1}^{K_b} b_{kl}\phi_k(t)$ and

$$\int_0^1 X_{il}(t)\beta(t)dt = \int_0^1 \mathbf{c}_{il}'\boldsymbol{\psi}_l^T(t)\boldsymbol{\phi}(t)\mathbf{b}_l\,dt = \mathbf{c}_{il}'\mathbf{J}_l\mathbf{b}_l,$$

where $\mathbf{c}_{il}' = (c_{i1l},\ldots,c_{iK_xl})^T$, $\mathbf{J}_l$ is a $K_x\times K_b$ dimensional matrix with the (k,m)th entry equal to $\int_0^1\psi_{kl}(t)\phi_m(t)dt$ Ramsay and Silverman (2005). We again induce smoothness on the

estimate of $\beta_l(t)$ by assuming $\mathbf{b}_l \sim N(0, \mathbf{D}_l)$, where $\mathbf{D}_l$ is the penalty matrix corresponding to the spline basis $\boldsymbol{\phi}(t)$ and the coefficient function $\beta_l(t)$.

Using notation analogous to the univariate case, the multivariate functional regression model (3) can be expressed in matrix format as

$$\begin{aligned} \mathbf{Y} \mid \mathbf{X}(\mathbf{t}) &\sim EF(\boldsymbol{\mu}, \boldsymbol{\gamma}) \\ g(\boldsymbol{\mu}) &= [1\ \mathbf{C}_1\mathbf{J}_1 ... \mathbf{C}_L\mathbf{J}_L\ \mathbf{Z}][\alpha\ \mathbf{b}_1 ... \mathbf{b}_L\ \boldsymbol{\gamma}]^T \\ \mathbf{b}_l &\sim \mathrm{N}(0, \mathbf{D}_l), l = 1, ..., L \end{aligned}$$

which is a mixed effect model with K_b random effects, $\mathbf{b}_l$, for each functional coefficient $\beta_l(t)$, and can be fit using standard mixed model software.

Note that we express each coefficient function in terms of the same spline basis; indeed, we typically use the truncated power series basis introduced in section 2.2 for each $\beta_l(t)$. However, different bases could be used for each function. Using $\phi_l(t) = \{\phi_{1l}(t), \phi_{2l}(t), ..., \phi_{K_b l}(t)\}$ as the basis for $\beta_l(t)$, the matrix J_l has $(k, m)^{th}$ entry equal to $\int_0^1 \psi_{kl}(t)\phi_{ml}(t)dt$; all other aspects of the multivariate regression model remain the same.

3.2 Multilevel extensions

Here, we briefly describe an extension of our method to a multilevel setting based on Crainiceanu et al. (2008).

Suppose for subject i we observe $[Y_i, \mathbf{Z}_i, \{W_{ij}(t), t \in [0, 1]\}]$, where Y_i is continuous or discrete, $\mathbf{Z}_i$ is a vector of covariates and $W_{ij}(t)$ is the observed functional regressor at visit $j = 1, 2, ..., J_i$. We assume that $W_{ij}(t)$ is a proxy for the true underlying subject-specific function $X_i(t)$, so that $W_{ij}(t) = \mu(t) + \eta_j(t) + X_i(t) + U_{ij}(t) + \epsilon_{ij}(t)$. Here $\mu(t)$ is the overall mean function, $\eta_j(t)$ is the visit-specific deviation from the overall mean, $X_i(t)$ is subject i's deviation from the visit-specific mean function, $U_{ij}(t)$ is the remaining subject- and visit-specific deviation for the subject specific mean, and $\epsilon_{ij}(t)$ is a white noise process with variance σ_ϵ^2. We further assume that $X_i(t), U_{ij}(t)$ and $\epsilon_{ij}(t)$ are uncorrelated to guarantee identifiability. We construct $\hat{\mu}(t) = \bar{W}_{\cdot\cdot}(t)$ and $\hat{\nu}_j(t) = \bar{W}_{\cdot\cdot}(t) = \bar{W}_{\cdot j}(t)$, where $\bar{W}_{\cdot\cdot}(t)$ is the mean taken over all subjects and visits and $\bar{W}_{\cdot j}(t)$ is the mean taken over all subjects at visit j. Assume these estimates have been subtracted, so that $W_{ij}(t) = X_i(t) + U_{ij}(t) + \epsilon_{ij}(t)$.

We use model (1) as our outcome model, so that the outcome Y_i depends on the subject-specific mean function $X_i(t)$. The multilevel approach proceeds analogously to the single-level approach. First, we express the subject-specific function $X_i(t)$ in terms of a parsimonious basis that captures most of the variability in the space spanned by the regressor

functions. Second, we express the coefficient function $\beta(t)$ using a truncated power series spline basis. Finally, we take advantage of the mixed models framework to construct a smooth estimate $\hat{\beta}(t)$.

We use Multilevel Functional Principal Components Analysis (MFPCA) Crainiceanu et al. (2008); Di et al. (2008) to construct parsimonious bases for $X_i(t), U_{ij}(t)$ based on the spectral decomposition of the covariance operators $K^X(s,t) = \text{Cov}[X_i(s), X_i(t)] = \sum_{k=1}^{\infty} \lambda_k^{(1)} \psi_k^{(1)}(s)\psi_k^{(1)}(t)$ and $K_T^U(s,t) = \text{Cov}[U_{ij}(s), U_{ij}(t)] = \sum_{l=1}^{\infty} \lambda_l^{(2)} \psi_l^{(2)}(s)\psi_l^{(2)}(t)$, where $\lambda_1^{(1)} \geq \lambda_2^{(1)} \geq \lambda_3^{(1)}...$ and $\lambda_1^{(2)} \geq \lambda_2^{(2)} \geq \lambda_3^{(2)}...$ are the ordered eigenvalues and $\boldsymbol{\psi}^{(1)}(\cdot) = \{\psi_i^{(1)}(\cdot) : i \in \mathbb{Z}^+\}$, $\boldsymbol{\psi}^{(2)}(\cdot) = \{\psi_i^{(2)}(\cdot) : i \in \mathbb{Z}^+\}$ are the corresponding orthonormal eigenfunctions. The Karhunen-Loéve decomposition is used to provide the finite series approximations $X_i(t) = \sum_{j=1}^{K_x} c_{ij}\psi_j^{(1)}(t)$ and $U_{ij}(t) = \sum_{l=1}^{L_x} \zeta_{ijl}\psi_l^{(2)}(t)$, where K_x and L_x are the truncation lags and $c_{ik} = \int_0^1 X_i(t)\psi_k^{(1)}(t)dt$, $\zeta_{ijk} = \int_0^1 U_{ij}(t)\psi_k^{(2)}(t)dt$ are the PC scores with $\text{E}[c_{ik}] = \text{E}[\zeta_{ijk}] = 0$, $\text{Var}[c_{ik}] = \lambda_k^{(1)}$, $\text{Var}[\zeta_{ijk}] = \lambda_k^{(2)}$, for every i, j, k. As in Crainiceanu et al. (2008), we estimate c_{ik}, ζ_{ijk} using the mixed model

$$W_{ij}(t) = \sum_{k=1}^{K_x} c_{ik}\psi_j^{(1)}(t) + \sum_{l=1}^{L} \zeta_{ijl}\psi_l^{(2)}(t) + \epsilon_{ij}(t) \quad (4)$$

$$c_{ik} \sim \text{N}[0, \lambda_k^{(1)}]; \zeta_{ijl} \sim \text{N}[0, \lambda_l^{(2)}]; \epsilon_{ij} \sim \text{N}[0, \sigma_\epsilon^2]. \quad (5)$$

Using the same notations as in the case of single-level regression the functional predictor becomes

$$\int_0^T X_i(s)\beta(s)ds = \int_0^T \mathbf{c}_i'[\boldsymbol{\psi}^{(1)}(s)]^T \boldsymbol{\phi}(s)\mathbf{b}\, ds = \mathbf{c}_i'\mathbf{J}_{\psi\phi}\mathbf{b}.$$

Thus, the outcome model is identical to model (3), with the only difference that $X_i(\cdot)$ are estimated using the MFPCA instead of the FPCA method. Penalized spline regression modeling is employed for modeling $\beta(t)$ and mixed model software is used.

This development is related to the one proposed in Crainiceanu et al. (2008). Specifically, the method in Crainiceanu et al. (2008) uses MFPCA to construct a parsimonious basis for $X_i(t)$ and uses a mixed model to estimate the PC loadings c_{ik}. Similarly to FPCR, the PC loadings are then treated as the regressors in a generalized linear model. In contrast, our method estimates $\beta(t)$ using a truncated power series spline basis and penalized regression to construct a smooth estimate $\hat{\beta}(t)$. This method is flexible and was found to be superior both in standard simulation settings and applications.

4 Sparse data

Our method also extends to the case where the functional regressor is measured sparsely at the subject level, but is dense across subjects. In this situation, we observe data of the form $[Y_i, \{W_i(t_{ij}) : t_{ij} \in [0,1]\}, \mathbf{Z}_i]$, $i = 1, \ldots, I$, $j = 1, \ldots, J_i$, where J_i are small but $\cup_{i=1}^{I} \left[\{t_{ij}\}_{j=1}^{J_i}\right]$ is dense in $[0,1]$. Here again, the $W_i(t)$ are measured-with-error proxies for the true $X_i(t)$ so that $W_i(t) = X_i(t) + \epsilon_i(t)$, where $\epsilon_i(t) \sim N(0, \sigma_\epsilon^2)$.

We use the following method, adapted from Di et al. (2008), to estimate the subject specific functional regressors based on a PC decomposition of the covariance operator $K^X(s,t)$. As indicated in section 2.1, we first use a fine grid of points on $[0,1]$ to obtain an undersmooth of the observed covariance matrix. Call the points in this grid $t_1, ..., t_S$, and for each subject let t_{ijs} be the point in this grid nearest to the observed point t_{ij}. The undersmoothed covariance operator can be estimated using $\hat{K}_1^W(r,s) = \sum_{i \in I(r,s)} \{W_i(t_{ijr}) - \overline{X}(t_r)\}\{W_i(t_{ijs}) - \overline{X}(t_s)\}/N(r,s)$, where $\overline{X}(t)$ is the mean of observed functions at t, $I(r,s)$ is the index of subjects with observed points corresponding to both t_r, t_s and $N(r,s)$ is the number of such subjects. We then smooth the off-diagonal elements of this undersmoothed covariance matrix to obtain $\hat{K}^X(r,s)$, the estimated covariance operator of the sparsely observed subject-specific functional regressors on our newly defined grid $t_1, ..., t_S$.

As before, we decompose $\hat{K}^X(r,s)$ so that $\hat{K}^X(r,s) = \sum_{k=1}^{\infty} \lambda_k \psi_k(r)\psi_k(s)$ where $\lambda_1 \geq \lambda_2 \geq \ldots$ are the non-increasing eigenvalues and $\boldsymbol{\psi}(\cdot) = \{\psi_k(\cdot) : k \in \mathbb{Z}^+\}$ are the corresponding orthonormal eigenfunctions. The function $X_i(t)$ is then approximated by $X_i(t) = \sum_{k=1}^{K_x} c_{ik}\psi_k(t)$, where K_x is the truncation lag and c_{ik} are the subject-specific PC loadings. Because subject-level data are sparse, numeric integration does not yield satisfactory estimates of the c_{ik}. Instead, in this case we propose the following mixed model to describe the observed data:

$$\begin{cases} W_i(t) &= \mu(t) + \sum_{k=1}^{K_x} c_{ik}\psi_k(t) + \epsilon_i(t) \\ c_{ik} &\sim \quad \mathrm{N}[0, \lambda_k]; \epsilon_{ij} \sim \mathrm{N}[0, \sigma_\epsilon^2] \end{cases}$$

where $\mu(t)$ is the mean function estimated across subjects. Here, the PC loadings are random effects and can be estimated using best linear unbiased predictions (BLUPs) or other standard inferential procedures. Note that the Gaussian assumption is convenient, but it could be relaxed.

Using the same notation as in other settings, the integral in the linear predictor of model (1) has the matrix representation $\int_0^T X_i(t)\beta(t)dt = \mathbf{c}_i'\mathbf{J}_{\psi\phi}\mathbf{b}$. Because of the sparseness of the subject-level data, it is often necessary to reduce the number of knots used in the spline basis

for $\beta(t)$ and the number of PCs used to explain the variability in the $X_i(t)$. In practice, we have found that $K_X = K_b = 10$ typically suffices. Penalized spline regression using mixed models can be used to fit this sparsely-sampled functional regression model.

5 Simulation

In this section, we pursue several simulation studies to explore the viability of our method in the univariate, multivariate, multilevel, and sparse functional regression settings. Where applicable, we compare our method to other existing approaches.

5.1 Univariate Simulations

We begin by investigating performance in the simplest situation - a single level, single functional regressor model, with a continuous outcome and no nonfunctional covariates. Consider the grid $\{t_g = \frac{g}{10} : g = 0, 1, \ldots, 100\}$ on the interval [0,10]. We generate scalar outcomes Y_i and regressor functions $X_i(t)$ from the following model

$$\begin{aligned} Y_i &= \frac{1}{G}\sum_{g=1}^{G} X_i(t_g)\beta(t_g) + \epsilon_i, \quad i = 1, \ldots, 200 \\ W_i(t_g) &= X_i(t_g) + \delta_i(t_g) \\ X_i(t_g) &= u_{i1} + u_{i2}t_g + \sum_{k=1}^{10}\left\{v_{ik1}\sin\left(\frac{2\pi k}{10}t_g\right) + v_{ik2}\cos\left(\frac{2\pi k}{10}t_g\right)\right\} \end{aligned} \tag{6}$$

where $\epsilon_i \sim \mathrm{N}[0, \sigma_\epsilon^2]$, $\delta_i(t_g) \sim \mathrm{N}[0, \sigma_X^2]$, $u_{i1} \sim \mathrm{N}[0, 25]$, $u_{i2} \sim \mathrm{N}[0, 0.04]$, and $v_{ik1}, v_{ik2} \sim \mathrm{N}[0, 1/k^2]$. For reference, Figure 1 displays a sample of 200 random functions $X_i(t)$ as well as the first six principal components estimated from the PC decomposition of the functions. This method of generating the regressor functions $X_i(t)$ is adapted from Muller and Stadtmuller (2005). The first principal components of the $X_i(t)$ capture a slope on t and sine and cosine functions with one, two, and three periods on the range of t. In generating the observed functions $W_i(t)$ we consider $\sigma_X^2 \in \{0, 1\}$, and in generating the observed outcomes Y_i, we consider $\sigma_\epsilon^2 \in \{0.5, 1\}$ and three true coefficient functions $\beta(\cdot)$, yielding 12 possible parameter combinations. The choices of the coefficient functions $\beta(\cdot)$ will be described below.

For each combination of the parameter values $\sigma_\epsilon^2, \sigma_X^2$, and $\beta(\cdot)$, we simulate 1200 datasets $[Y_i, W_i(t_g) : i = 1, \ldots, 200]$. We compare three alternative approaches to estimating $\beta(\cdot)$ to our approach as described in Section 2. Performance in estimating $\beta(\cdot)$ is compared by

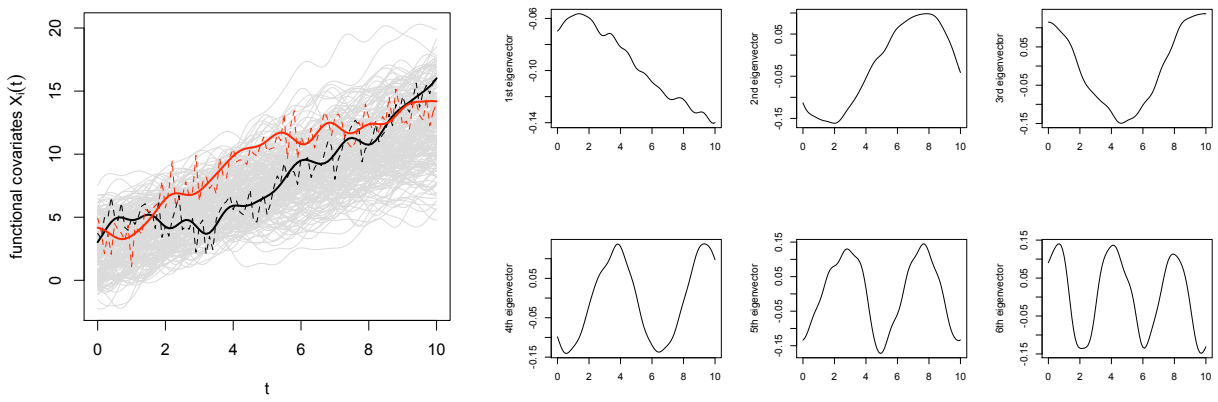

Figure 1: The left panel displays a sample of 200 random functions generated from equation (6), highlighting two examples of the function measured with no error (solid) and with measurement error $\sigma_X^2 = 1$ (dashed). The right panel displays plots of the first 6 estimated principal components.

calculating the average mean square error (AMSE) over the 1200 samples as

$$\text{AMSE}(\hat{\beta}(\cdot)) = \frac{1}{1200}\sum_{r=1}^{1200}\left[\frac{1}{G}\sum_{g=1}^{G}\left\{\hat{\beta}_r(t_g) - \beta(t_g)\right\}^2\right],$$

where $\hat{\beta}_r(\cdot)$ is coefficient function from the r^{th} simulated data set.

The first method for estimating $\beta(\cdot)$ is principal components regression (PCR). Let $\mathbf{X}$ be the $200 \times G$ matrix with i^{th} row $(\mathbf{X}_i(t_1), \ldots, X_i(t_G))$ and calculate the singular value decomposition $\mathbf{UDV}^{\text{T}}$ of $\mathbf{X}$. In PCR, the scalar outcomes Y are regressed on $\mathbf{V}_{\text{A}}$, the $200 \times A$ matrix containing the first A columns of $\mathbf{V}$, which are also referred to as the first A principal components of $\mathbf{X}$. We consider two commonly used approaches for selecting A: cross validation (PCR-CV) and percent variance explained (PCR-PVE) with a 99% threshold. For PCR-CV we implement the leave-one-out cross validation procedure to select the number of principal components A for which the prediction sum of squares criterion $\sum_{i=1}^{n}(Y_i - \hat{Y}_{-i})^2$ is minimized. Here $\hat{Y}_{-i}$ is the predicted value for the i^{th} data point obtained from fitting the PCR model to the data with the i^{th} observation deleted. Though this procedure is computationally intensive, we implement a faster alternative formulation for the statistic for a linear model, namely

$$\sum_{i=1}^{n}\left(\frac{Y_i - \hat{Y}_i}{1 - H_{ii}}\right)^2,$$

where H_{ii} is the i^{th} diagonal element of the regression projection matrix $H = \mathbf{V}_{\mathrm{A}}(\mathbf{V}'_{\mathrm{A}}\mathbf{V}_{\mathrm{A}})^{-1}\mathbf{V}_{\mathrm{A}}$ and $\hat{Y}_i$ is the i^{th} fitted value. For PCR-PVE with a 99% threshold, we select the value of A satisfying

$$A = \min\left\{a : \frac{\lambda_1 + \cdots + \lambda_a}{\lambda_1 + \cdots + \lambda_G} \leq 0.99\right\},$$

where λ_a is the eigenvalue corresponding to the a^{th} principal component of $\mathbf{X}$. Thus we interpret A as the minimal number of principal components needed to explain 99% of the total variation in the discretized versions of the random functions $X_i(t)$.

The second method FPCR$_R$ (Reiss and Ogden (2007)) first projects the random functions $X_i(t)$ onto a B-spline basis and then performs a principal components analysis on the projection XB. A penalized regression model is then fit to find ξ that minimizes the criterion

$$||Y - \mathbf{X}\mathbf{B}\mathbf{V}_{\mathrm{A}}\boldsymbol{\beta}||^2 + \lambda\xi^T\mathbf{V}_{\mathrm{A}}^{\mathrm{T}}\mathrm{P}^{\mathrm{T}}\mathrm{P}\mathbf{V}_{\mathrm{A}}\xi,$$

where $\mathbf{V}_{\mathrm{A}}$ is the first A columns of $\mathbf{V}$ from the singular value decomposition $\mathbf{XB} = \mathbf{UDV}^{\mathrm{T}}$. Given a particular value of A, the smoothing parameter λ may be selected either through GCV or by representing the penalized regression in a LMM framework and using the REML estimate. The number of principal components A is selected by multi-fold cross validation. We implement this method using code provided by the authors, which utilizes the REML estimate, a cubic B-splines basis with 40 equally spaced internal knots, and selects the number of principal components A using 8-fold cross validation. The candidates for A were 1-10, 12, and 15-40 at intervals of 5.

Finally, we implement the method SPCR-GCV (Cardot et al. (2003)), using code provided by the authors. This approach first computes the PCR estimate $\hat{\beta}_{\mathrm{PCR}}(t)$ using the first K principal components and then smoothes the resulting function using penalized splines. In this approach, both the dimension K of the principal components basis and the smoothing parameter ρ are selected using generalized cross validation (GCV). The number of knots of the B-spline basis and the degree of the spline functions were fixed at 20 and 4, respectively. We consider the same candidates for K as were used to select the number of principal components (A) in the implementation of FPCR$_R$ described above, and the candidates for ρ were 10^{-8} to 10^{-7} by intervals of 10^{-8}, 10^{-7} to 10^{-6} by intervals of 10^{-7}, and 10^{-6} to 10^{-5} by intervals of 10^{-6}. We selected this range in order to contain the minimum GCV values for each of the three true $\beta(\cdot)$ functions. We note that without this manual tuning the method fails to work well.

The true coefficient functions we consider in our simulations are $\beta_1(t) = \sin(\pi t/5)$, $\beta_2(t) = \sqrt{t}$, and $\beta_3(t) = -p(t \mid 2, 0.3) + 3p(t \mid 5, 0.4) + p(t \mid 7.5, 0.5)$, where $phi(\cdot \mid \mu, \sigma)$ is the normal

density with mean μ and standard deviation σ. The function $\beta_1(t)$ was selected because it is one of the functions used to generate the random functions $X_i(t)$, and is expected to favor methods that use the principal components basis for $\beta(t)$. Both PCR methods (CV and PVE) and FPCR$_R$ use the principal components as a basis for the unknown $\beta(\cdot)$. The second coefficient function was chosen as an arbitrary and realistic smooth coefficient function. The third has spikes at places where the variability in the $X_i(t)$ is low, meaning that the peaks will be very hard to detect with small sample sizes; we expect it will be difficult to estimate for all of the approaches used here.

Method	$\beta_1(\cdot)$		$\beta_2(\cdot)$		$\beta_3(\cdot)$	
	$\sigma^2_\epsilon = 0.5$	$\sigma^2_\epsilon = 1$	$\sigma^2_\epsilon = 0.5$	$\sigma^2_\epsilon = 1$	$\sigma^2_\epsilon = 0.5$	$\sigma^2_\epsilon = 1$
PFR						
σ^2_X = 0	0.0023	0.0037	0.003	0.004	0.188	0.234
σ^2_X = 1	0.0032	0.0042	0.0068	0.007	0.271	0.283
FPCR$_R$						
σ^2_X = 0	0.0019	0.0024	0.0238	0.0301	0.15	0.193
σ^2_X = 1	0.0054	0.0062	0.033	0.0366	0.255	0.266
SPCR-GCV						
σ^2_X = 0	0.0053	0.0091	0.0061	0.0098	0.157	0.177
σ^2_X = 1	0.0076	0.0103	0.0104	0.0126	0.247	0.259
PCR-PVE						
σ^2_X = 0	0.002	0.0031	0.017	0.0181	0.289	0.290
σ^2_X = 1	0.389	0.5850	0.429	0.626	0.581	0.778
PCR-CV						
σ^2_X = 0	0.058	0.1130	0.0936	0.172	0.381	0.615
σ^2_X = 1	0.0108	0.0128	0.0329	0.0363	0.306	0.313

Table 1: Average MSE over the 1200 repetitions for each combination of the true coefficient function $\beta(t)$, the measurement error variance σ^2_X and the outcome variance σ^2_ϵ.

Table 1 compares the AMSE for each set of the parameters across approaches. We first note that both our approach and SPCR-GCV have smaller AMSE than the PCR-CV for every parameter combination, and the FPCR$_R$ method has smaller AMSE than PCR-CV in all cases except for β_2 with $\sigma^2_X = 1$, in which case the two yield very similar results. The PCR-PVE method has poor performance for $\sigma^2_X \neq 0$ compared to all other methods.

When there is no measurement error, its performance is similar to both our approach and the FPCR$_R$ approach and superior to the SPCR-GCV and CV methods for β_1; for β_2 and β_3, PCR-PVE is worse than nearly all other methods except CV, with the exception of FPCR$_R$ for $\beta_2(\cdot)$ with σ_ϵ^2.

The function β_1 was selected because it is a basis function for the $X_i(t)$, so the methods that use the principal components as a basis for $\beta(t)$ (PCR-CV, PCR-PVE, FPCR$_R$, and SPCR-GCV) are expected to perform well. However, our method performs only slightly worse for β_1 than both the FPCR$_R$ and PVE methods when $\sigma_X^2 = 0$ and performs slightly better when $\sigma_X^2 = 1$; our method also has less than half the AMSE as the SPCR-GCV and PCR-CV approaches, with and without measurement error on the $X_i(t)$.

For the smooth β_2, our approach performs much better than SPCR-GCV which, in turn, performed much better than than the other approaches. As expected, none of the methods perform well for the third true coefficient function β_3; The FPCR$_R$ and SPCR-GCV methods provides the closest estimates, with our method performing slightly worse. In Figure 2 we

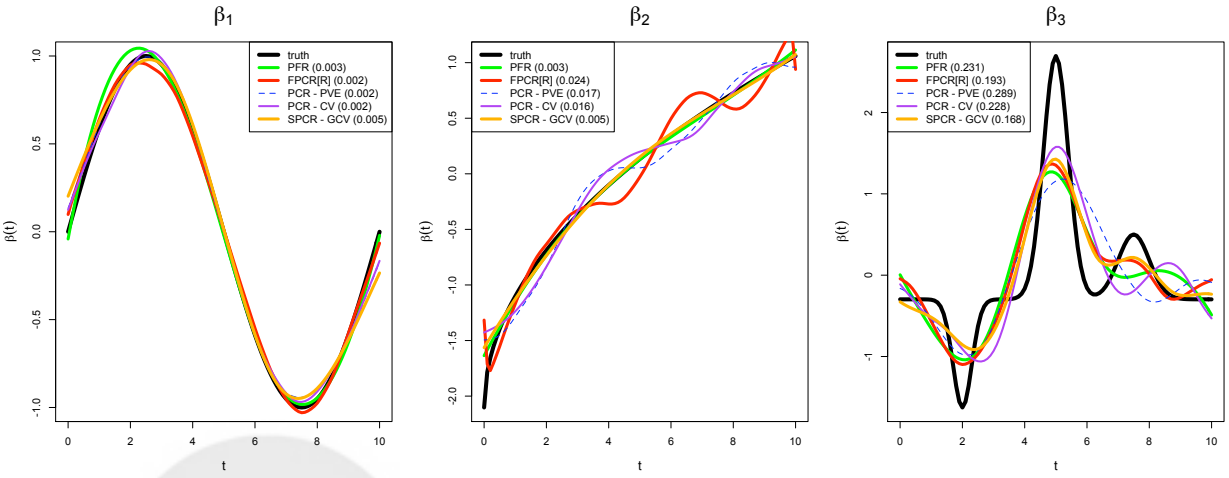

Figure 2: For the simulation with $\sigma_X^2 = 0$ and $\sigma_\epsilon^2 = 1$, we plot the estimated beta functions $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ from each method that have the median MSE.

select the estimated beta functions from each approach that have the median MSE for the case where $\sigma_X^2 = 0$ and $\sigma_\epsilon^2 = 1$. This plot reiterates the comparable performance across methods for β_1, the superiority of our approach as well as SPCR-GCV for the smooth β_2, and the relatively poor performance across all methods for β_3.

Another consideration in fitting functional models is computation time, particularly as the sample size n increases. To compare computation time in our approach to that in the

FPCR$_R$ and SPCR-GCV approaches, we examined the case where $\sigma_X^2 = 0$ and $\sigma_\epsilon^2 = 1$. To investigate how much computation time increases as sample size increases, we considered $n =$ 100, 200, 400, and 2000. For each n, we generated a single dataset $[Y_i, W_i(t_g) : i = 1, \ldots, n]$ with true coefficient function β_1 and fit each model 10 times. The average computation

n	PFR	SPCR-GCV	FPCR$_R$
100	0.111	2.451	16.720
200	0.126	4.536	18.545
400	0.157	13.330	26.070
2000	0.390	231.214	57.469

Table 2: Mean computation time (seconds) over 10 model fits by sample size and regression approach, for $\sigma_X^2 = 0$, $\sigma_\epsilon^2 = 1$, and $\beta(t) = \beta_1(t)$.

time for a single fit across the three methods is displayed in Table 2. The driver behind increasing computation times as sample size increases is implementation of a cross validation or generalized cross validation procedure. In the FPCR$_R$ method, 8-fold cross validation selects the number of principal components A. Though generalized cross validation reduces the computational burden of cross validation, the SPCR-GCV approach has a nested GCV procedure, leading to a large increase in computation time as the sample size n doubles. It should not be surprising that such computational problems would snowballed in more complex settings. In fact, no competing method was generalized to the more complex settings considered in this paper. The computational issues pointed out above are probably the main reason for this.

5.1.1 Confidence Intervals

We evaluate the performance of 95% pointwise confidence intervals for $\hat{\beta}(t)$ for our approach, using the methodology described in section 2.3 for each of the three true $\beta(t)$. For each point t_g along the range $[0, 10]$, let (l_g, u_g) denote the estimated 95% confidence interval about $\hat{\beta}(t_g)$. We compute the proportion of times during the 1200 iterations of the simulation that the calculated interval (l_g, u_g) contains the truth $\beta(t_g)$. These proportions are displayed in Figure 3 for the cases where $\sigma_\epsilon^2 = 0.5$ without measurement error on the $X_i(t)$; taking $\sigma_\epsilon^2 = 1$ yields similar confidence interval coverage probabilities.

The observed coverage proportions are often significantly below the nominal coverage probabilities. A likely reason for this is the variability in estimating the subject-specific PC

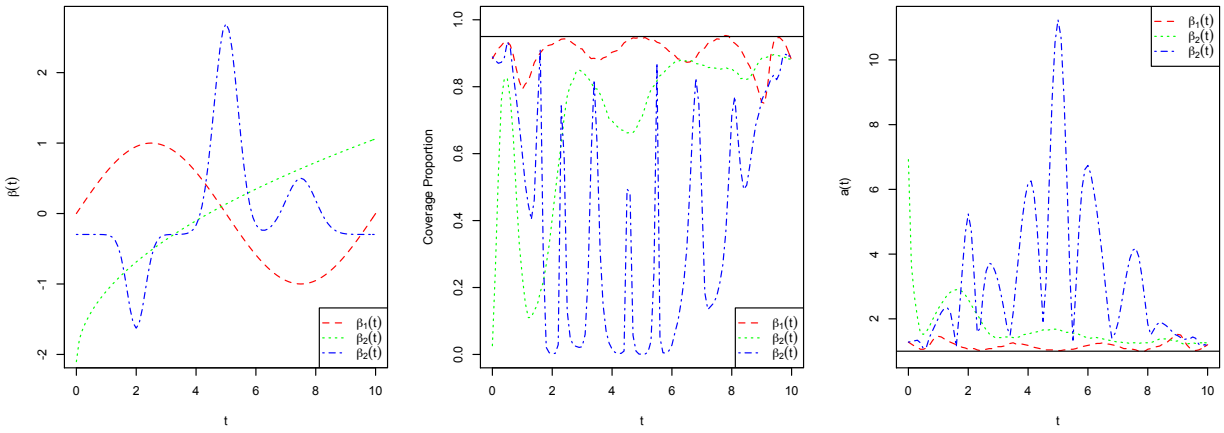

Figure 3: For the simulation with $\sigma_\epsilon^2 = 0.5$, a plot of the point-wise coverage probabilities for each true $\beta(t)$ and the correction factor $a(t)$.

loadings is not accounted for in the construction of the interval. Another possibility is that the functional regressors are not highly variable in some regions, making estimation of $\beta(t)$ difficult (see especially $\beta_3(t)$, and $\beta_2(t)$ for $t < .5$). To examine the underperformance of the confidence intervals, for each $\beta(t)$ in our simulation we calculate a correction factor $a(t)$ such that the interval $\hat{\beta}(t_0) \pm 1.96\, a(t) \widehat{\text{sd}}\{\hat{\beta}(t)\}$ achieves 95% coverage. A plot of $a(t)$ is included in Figure 3. The correction factor $a(t)$ is quite large for $\beta_3(t)$, but not unreasonable for $\beta_1(t), \beta_2(t)$. A similar approach could be used in an application: simulate data from $\hat{\beta}(t)$ and calculate $a(t)$ needed to achieve 95% coverage.

Figure 4 displays plots of the true $\beta(t)$, along with plots of various empirical quantiles of the estimated confidence intervals over the 1200 iterations for the case where $\sigma_X^2 = 0$ and $\sigma_\epsilon^2 = 1$. For β_1, confidence intervals performed the best, achieving the nominal coverage over some subsets of the range $[0, 10]$ of t, and with coverage between 78-95% across the entire range when there was no measurement error on the $X_i(t)$ and coverage between 71-96% with measurement error. For β_2, as can be seen in Figure 2, all approaches struggle with estimation at the lower range of t, and the confidence intervals have corresponding difficulty with coverage at these values. As larger values of t, coverage improves, but remains consistently below the nominal rate. For β_3, estimation in our approach is unable to capture the bumpy shape and as a result coverage is quite poor for most of the range of t.

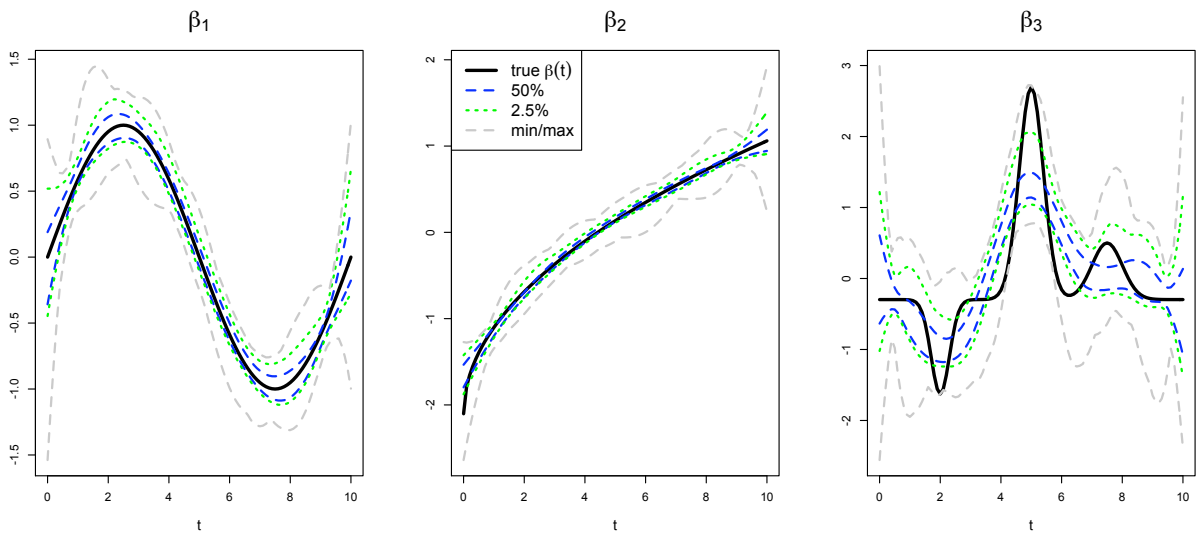

Figure 4: For the simulation with $\sigma_X^2 = 0$ and $\sigma_\epsilon^2 = 1$, the median pointwise confidence intervals, as well as the lower and upper 2.5%, and minimum and maximum.

5.2 Multivariate Simulations

In this section, we pursue a simulation exercise to study the model presented in section 3.1.

We generate samples from the model

$$\begin{aligned} Y_i &= \int_0^{10} X_{i1}(t)\beta_1(t)dt + \int_0^{10} X_{i2}(t)\beta_2(t)dt + \epsilon_i, \quad i = 1, \ldots, 200 \\ W_{i1}(t) &= X_{i1}(t) + \delta_{i1}(t);\ W_{i2}(t) = X_{i2}(t) + \delta_{i2}(t) \\ X_{i1}(t) &= u_{i1} + u_{i2}t + \sum_{k=1}^{10} \left\{ v_{ik1} \sin\left(\frac{2\pi k}{10}t\right) + v_{ik2} \cos\left(\frac{2\pi k}{10}t\right) \right\} \\ X_{i2}(t) &= a + .2(t-b)^2 + c\cos(\frac{2\pi t}{d}) \end{aligned}$$

where $\epsilon_i \sim \mathrm{N}[0, \sigma_\epsilon^2]$, $\delta_{li}(t_g) \sim \mathrm{N}[0, \sigma_X^2]$, $u_{i1} \sim \mathrm{N}[0, 25]$, $u_{i2} \sim \mathrm{N}[0, 0.04]$, $v_{ik1}, v_{ik2} \sim \mathrm{N}[0, 1/k^2]$, $a \sim U[0, 5]$, $b \sim N[5, \sigma = .5]$, $c \sim N[1, 1]$, and $d \sim U[4, 6]$. We assume $I = 200$ subjects, and select $\beta_1(t) = \sin(t)$ and $\beta_2(t) = \sqrt{t}$. The first functional regressor is generated as in Equation 6; the second functional regressor consists of a random intercept, a parabola with randomly shifted minimum, and a cosine term with random period and amplitude. Figure 5 provides samples of the random functions used in our current simulation exercise.

We simulated 1000 such datasets, and used the method given in section 3.1 to fit the multivariate functional regression. We are unaware of other methods for fitting this model; however, for comparison, we implemented a straightforward extension of the PCR-PVE

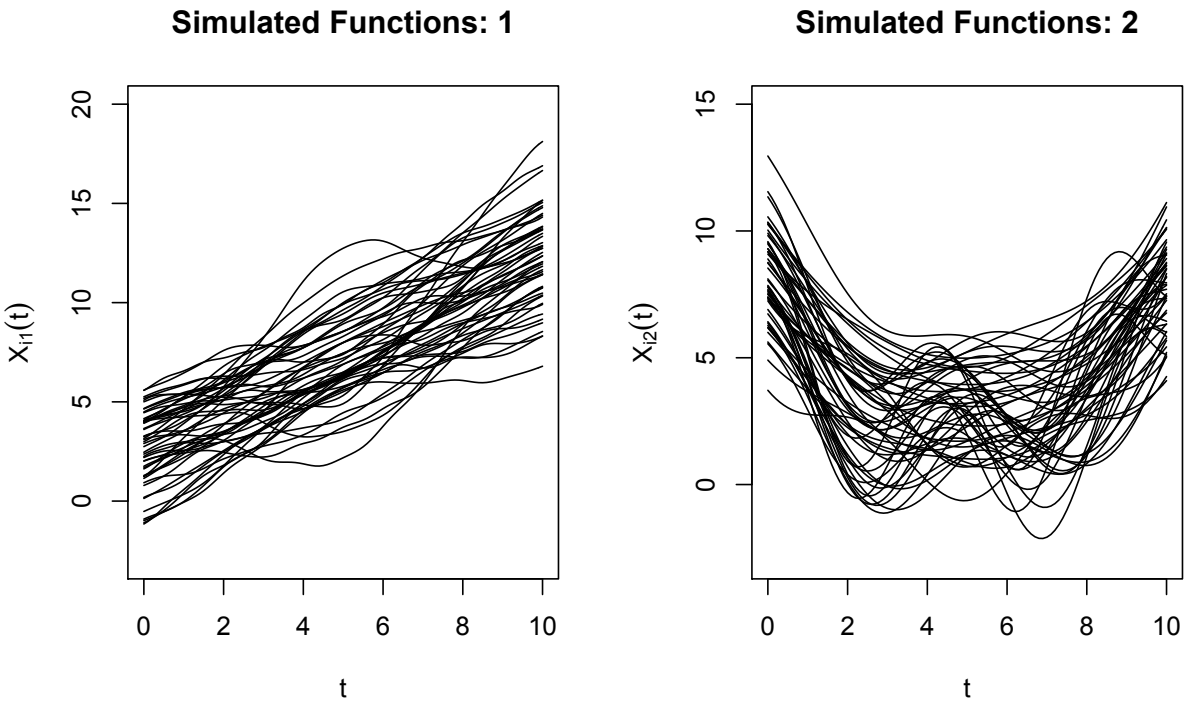

Figure 5: A sample of random functions generated according to different schemes. In the left panel, functions are combinations of random intercepts, slopes, and several sine and cosine terms. In the right panel, functions are combinations of random intercepts, parabolas shifted horizontally by random amounts, and cosines of random period and amplitude.

method used in the univariate case. That is, we regressed on the principal component loadings for each of the regressor functions (decomposed separately), choosing the number of loadings for each regressor function based on the percent of the variance explained by the eigenfunctions. Note that, similarly to section 5.1, the $\beta_1(t)$ is exactly a principal component of the $X_{i1}(t)$, which favors PCR-PVE, while $\beta_2(t)$ is an arbitrary smooth function.

Table 3 shows the results of the simulation study of the multivariate functional regression model. Not surprisingly, the average MSEs are higher for our method in this simulation than in the univariate simulations: without increasing the number of subjects, we have added complexity to the model. Despite this, our penalized spline approach continues to perform well in the multivariate setting, for both noiseless and noisy observations of functional regressors. We again see that the PCR-PVE method performs well when the coefficient function is exactly an early principal component of the corresponding regressor functions, less well

Method	True $\beta_1(\cdot)$		True $\beta_2(\cdot)$	
	$\sigma^2_\epsilon = .5$	$\sigma^2_\epsilon = 1$	$\sigma^2_\epsilon = .5$	$\sigma^2_\epsilon = 1$
PFR				
σ^2_X = 0	0.0063	0.0092	0.0053	0.0061
σ^2_X = 1	0.0269	0.0278	0.0205	0.0211
PCR-PVE				
σ^2_X = 0	0.0047	0.0060	0.0234	0.0240
σ^2_X = 1	0.4530	0.483	0.7116	0.7563

Table 3: Mean MSE over the 1000 simulated multivariate functional regression models for the method presented in this manuscript and an adapted PCR-PVE method.

when the coefficient function is an arbitrary smooth function and considerably worse in the presence of measurement error.

5.3 Multilevel Simulations

Next we pursue a brief simulation exercise to examine the performance of our proposed method in the multilevel setting (Section 3.2).

We generate samples from the model

$$\begin{aligned} Y_i &= \int_0^{10} X_i(t)\beta(t)dt + \epsilon_i, \quad i = 1, \ldots, 200 \\ W_{ij}(t) &= X_i(t) + U_{ij}(t) + \delta_{ij}(t), \quad j = 1, \ldots, 3 \\ X_{i1}(t) &= u_{i1} + u_{i2}t + \sum_{k=1}^{10} \left\{ v_{ik1} \sin\left(\frac{2\pi k}{10}t\right) + v_{ik2} \cos\left(\frac{2\pi k}{10}t\right) \right\} \\ U_{ij}(t) &= a \cdot f_1(t) + b \cdot f_2(t) + c \cdot f_3(t) \end{aligned}$$

where $\epsilon_i \sim \mathrm{N}[0, \sigma^2_\epsilon]$, $\delta_{li}(t_g) \sim \mathrm{N}[0, \sigma^2_X]$, $u_{i1} \sim \mathrm{N}[0, 25]$, $u_{i2} \sim \mathrm{N}[0, 0.04]$, $v_{ik1}, v_{ik2} \sim \mathrm{N}[0, 1/k^2]$, $a \sim N[0, \sigma^2 = 2]$, $b \sim N[0, \sigma^2 = 1]$, and $c \sim N[0, \sigma^2 = .5]$. Further, the components used in the construction of the $U_{ij}(t)$ are given by

$$f_1(t) = \frac{1}{\sqrt{10}};\ f_2(t) = \sqrt{\frac{3}{10}}\left(\frac{t}{5} - 1\right);\ f_3(t) = \sqrt{\frac{5}{10}}\left\{6\left(\frac{t}{10}\right)^2 - 6\left(\frac{t}{10}\right) + 1\right\}.$$

Again, we choose $\beta_1(t) = \sin(t)$ and $\beta_2(t) = \sqrt{t}$.

We generate 100 such data sets and fit the resulting models using the extension detailed in Section 3.2. Here (as in Di et al. (2008)), we estimate the subject-specific PC loadings

in model (4) using Markov chain Monte Carlo because $\boldsymbol{\psi}_j^{(1)}$ and $\boldsymbol{\psi}_j^{(2)}$ are not mutually orthogonal. We compare the penalized functional regression presented here to the functional regression method described in Di et al. (2008), which is an extension of PCR-PVE.

Method	True $\beta_1(\cdot)$		True $\beta_2(\cdot)$	
	$\sigma_\epsilon^2 = .5$	$\sigma_\epsilon^2 = 1$	$\sigma_\epsilon^2 = .5$	$\sigma_\epsilon^2 = 1$
PFR				
σ_X^2 = 0	0.0344	0.0663	0.0225	0.0207
σ_X^2 = 1	0.1301	0.1356	0.0241	0.0251
PCR-PVE				
σ_X^2 = 0	0.0086	0.0094	0.0751	0.0761
σ_X^2 = 1	0.0133	0.0120	0.0864	0.0877

Table 4: Mean MSE over the 1000 simulated multivariate functional regression models for the method presented in this manuscript and an adapted PCR-PVE method.

Table 4 shows the results of this simulation. As before, when the coefficient function is an arbitrary smooth function, the PFR method performs several times better than the PCR-PVE method. Also as before, the PCR-PVE approach outperforms the PFR method when the the coefficient function is taken to be an early principal component. Unlike before, this advantage remains in the presence of measurement error. The mostly likely reason for this is that the PCR-PVE method described in Di et al. (2008) uses a smoothed covariance matrix to estimate the PCs.

5.4 Sparse Data Simulations

Our final simulations test the extension of our method to the case where the functional regressor is sparsely observed at the subject level but densely observed over subjects, as described in Section 4.

We generate samples from the model

$$\begin{aligned} Y_i &= \int_0^{10} X_i(t)\beta(t)dt + \epsilon_i, \quad i = 1, \ldots, 200 \\ W_i(t) &= X_i(t) + \delta_i(t) \\ X_{i1}(t) &= u_{i1} + u_{i2}t + \sum_{k=1}^{10} \left\{ v_{ik1} \sin\left(\frac{2\pi k}{10}t\right) + v_{ik2} \cos\left(\frac{2\pi k}{10}t\right) \right\} \end{aligned}$$

where $\epsilon_i \sim \mathrm{N}[0, \sigma_\epsilon^2]$, $\delta_{li}(t_g) \sim \mathrm{N}[0, \sigma_X^2]$, $u_{i1} \sim \mathrm{N}[0, 25]$, $u_{i2} \sim \mathrm{N}[0, 0.04]$, and $v_{ik1}, v_{ik2} \sim \mathrm{N}[0, 1/k^2]$. Sparseness at the subject level is introduced by uniformly sampling 10 points in T independently for each subject, so that for each subject we observe $[Y_i, \{W_i(t_{ij}) : t_{ij} \in [0, 10]\}]$, $i = 1, \ldots, 500$, $j = 1, \ldots, 10$; note however that the sampling takes place after the outcome is generated.

We simulate 1000 data sets in this way, and use the extension described in Section 4 to fit the functional regression model for sparsely observed subject-level data. For our simulations, we estimate the covariance operator $K^W(s, t)$ on the full grid T rather than a subset thereof, but we note that our code allows one to use a smaller grid to undersmooth the covariance operator. Figure 6 shows a sample of sparsely observed functions, as well as the estimated function based on the PC decomposition of $K^W(s, t)$.

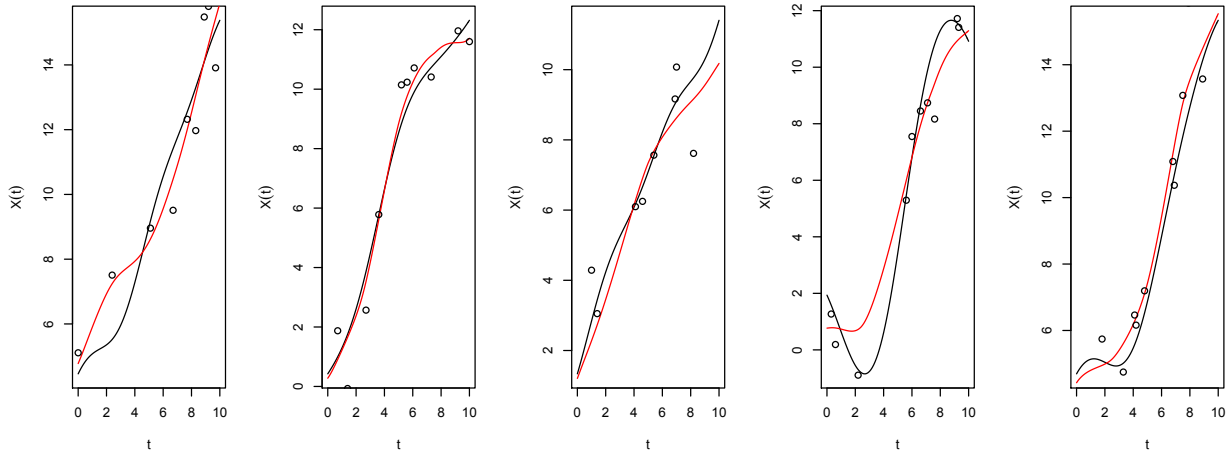

Figure 6: A sample of sparsely observed functions, measured with error. For each panel, the black curve represents the true $X_i(t)$, the black points are observed points (the $X_i(t_{ij})$, and the red curve is the estimated function.

Table 5 gives the results of the sparsely observed functional regression simulation. Because of the presence of a few very large MSEs, we include both the average and the median MSE; also, we indicate whether σ_ϵ^2 is known or unknown.

Somewhat paradoxically, the presence of measurement error seems to dramatically improve the estimation of $\beta(t)$. This stems from the estimation of the $X_i(t)$: in the case of no or little measurement error, the functions are systematically over- or underestimated at various regions of $[0, 10]$, leading to similar errors across subjects. In turn, this dramatically reduces the ability to estimate $\beta(t)$. In the case of higher measurement error, the across-subject errors are less biased and the estimate of $\beta(t)$ are better.

Not surprisingly, the median MSE is higher in the sparse data case than in the univariate regression simulation of Section 5.1. However, we are typically able estimate $\beta(t)$ fairly accurately with comparatively little information for each subject by borrowing information across subjects to estimate the $X_i(t)$.

Method	True $\beta_1(\cdot)$		True $\beta_2(\cdot)$	
	$\sigma_\epsilon^2 = 0.5$	$\sigma_\epsilon^2 = 1$	$\sigma_\epsilon^2 = 0.5$	$\sigma_\epsilon^2 = 1$
σ_X^2 unknown				
σ_X^2 = 0	3751	3785	26230	25410
	(.2596)	(.1974)	(.0242)	(.0215)
σ_X^2 = 1	1.942	1.9810	0.0991	0.1212
	(.0856)	(.0853)	(.0175)	(.0175)
σ_X^2 known				
σ_X^2 = 0	1.657	1.690	.5774	.6787
	(.0366)	(.0354)	(.0138)	(.0138)
σ_X^2 = 1	1.574	1.331	.0990	.0886
	(.0753)	(.0755)	(.0170)	(.0170)

Table 5: Mean MSE over the 1000 repetitions for each combination of the true coefficient function $\beta(t)$, the measurement error variance σ_X^2 and the outcome variance σ_ϵ^2. Median MSE is given in parentheses.

6 Application to DTI Tractography

Our application is to a study comparing the cerebral white matter tracts of multiple sclerosis patients to the tracts of controls. White matter tracts consist of axons, the long projections of nerve cells that carry electrical signals, that are surrounded by a fatty insulation called myelin. The myelin sheath allows an axon in a white matter tract to transmit signals at a much faster rate than is possible in a non-myelinated axon. Multiple sclerosis is a demyelinating autoimmune disease that causes lesions in the white-matter tracts of affected individual and results in severe disability.

Diffusion tensor imaging (DTI) tractography is a magnetic resonance imaging (MRI) technique that allows the study of white-matter tracts by measuring the diffusivity of water in the brain: in white-matter tracts, water diffuses anisotropically in the direction of the

tract, while elsewhere water diffuses isotropically. Using measurements of diffusivity along several gradients, DTI can provide relatively detailed images of white-matter anatomy in the brain Basser et al. (1994, 2000); LeBihan et al. (2001); Mori and Barker (1999).

For each white-matter tract, DTI provides us several measures describing the diffusivity of water. One example of these measures is parallel diffusivity, which is the diffusivity along the principal axis of the tract. Parallel diffusivity is recorded at many locations along the tract, so that for each tract we have a continuous profile or function. Figure 7 shows the parallel diffusivity profile for a single tract, separated into cases and controls.

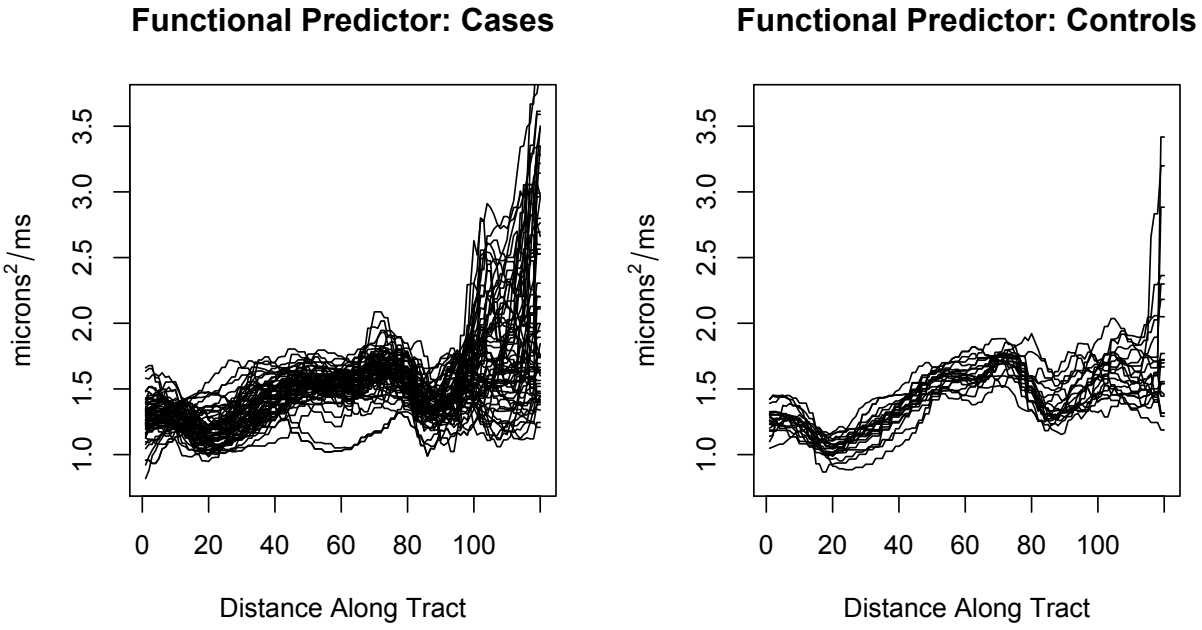

Figure 7: Display of the parallel diffusivity profile for the left intracranial cortico-spinal tract, separated by MS status.

Our study consists of 20 controls and 65 cases, for whom we have a full DTI scan at baseline. Here, we focus on parallel diffusivity profiles as a way to classify subjects as cases or controls. Specifically, we take as our functional predictor the parallel diffusivity profile of the left intracranial cortico-spinal tract. Our first approach to this problem builds intuition: we bin the parallel diffusivity profiles and regress on the bin means, keeping those that are significantly related to the MS status. While straightforward, we recall that this is equivalent to constraining $\beta(t)$ in a functional regression model to be a step function. We compare this to the penalized functional regression model presented in this manuscript.

The far-left panel of Figure 8 shows the estimates $\hat{\beta}(t)$ resulting from the two approaches. Both approaches emphasize the same two regions of the tract as important for distinguishing cases from controls, and give similar weights to these regions. Thus, those individuals whose parallel diffusivity profile is above average between distances 20 and 40 are more likely to be MS patients. Similarly, those individuals whose parallel diffusivity profile is above average between distances 50 and 65 are less likely to be MS patients. Moreover, the middle-left panel of this compares the predictive ability of the bin-mean and PFR methods via their respective ROC curves.

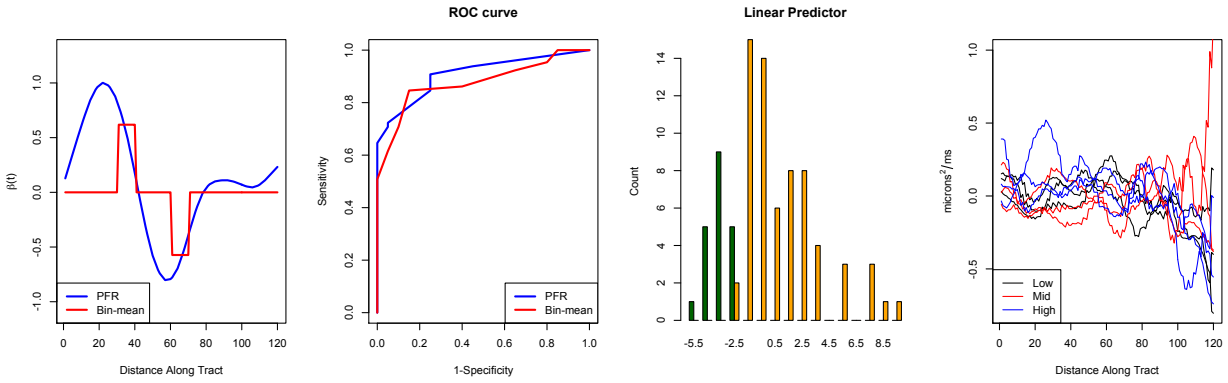

Figure 8: The far-left panel shows the estimated $\beta(t)$from the PFR and bin-mean approaches. The middle-left panel shows the ROC curves generated by these approaches. The middle-right panel shows the distribution of the linear predictor $\int_0^1 \beta(t)X_i(t)\,dt$ for the PFR method (cases orange, controls green). The far-right panel shows the tract profiles with the lowest (black), middle (red), and highest (blue) linear predictors (Note that the tract profiles have been de-meaned).

For each subject, we also compute the linear predictor $\int \mathbf{X}_i(t)\beta(t)dt$ from the PFR method; the middle-right panel of Figure 8 shows the distribution of these quantities for both cases and controls. As anticipated, $\int \mathbf{X}_i(t)\beta(t)dt$ provides a reasonable quantity for distinguishing cases from controls based on the tract profile. The far-right panel of Figure 8 compares the tract profile resulting in the lowest three, the middle three and the highest three linear predictors. We note that the tract profiles in this panel are $X_i(t) - \mu(t)$, where $\mu(t)$ is the overall mean profile. Thus, profiles with a low linear predictor will tend to be below zero between distances 20-40 and above average between distances 50-65, and conversely for profiles with high linear predictors.

7 Discussion

By combining several well-known techniques in FDA, we have developed a method for generalized functional regression with the following properties: *i*. flexibly estimates $\beta(t)$; *ii*. is applicable in cases of measurement error, multilevel observations, and sparse data; and *iii*. compares favorably with existing methods in simulation studies. Although it builds on existing work, this method is conceptually new in that we estimate the regressor functions $X_i(t)$ using a PC decomposition, which allows the use of our method when the $X_i(t)$ are poorly observed (measurement error, sparse observation) or unobserved (multilevel). Further, by expressing our method in terms of a GLMM, we take advantage of well-researched and computationally efficient machinery for fitting the model.

We tested our method tested in each of the settings we describe, with good results. We note that our simulation highlighted a case in which our method (as well as the others we examined) performed poorly. It is inherently difficult to detect peaks in $\beta(t)$ when those peaks occur in areas of low variability in the $X_i(t)$. Another interesting case is that of sparsely observed functions in the absence of measurement error. When the measurement error variance is treated as unknown, it is estimated with bias and can result in poor estimates of $\beta(t)$; however, fixing $\sigma_\epsilon^2 = .05$ generally resolves this issue. We note that another possible solution could be fully Bayesian treatment of the functional regression.

Several directions for future work are apparent. Handling several functional regressors, especially when those regressors are correlated, will be important as larger and larger data sets become available, as will developing new methods for multilevel functional data. More generally, examining the effectiveness of functional methods compared to less sophisticated techniques is necessary to establish the practical justification for these methods.

8 Acknowledgements

Crainiceanu's, Goldsmith's, and Caffo's research was supported by Award Number R01NS060910 from the National Institute Of Neurological Disorders And Stroke. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute Of Neurological Disorders And Stroke or the National Institutes of Health. This research was partially supported by the Intramural Research Program of the National Institute of Neurological Disorders and Stroke. We also thank the National Multiple Sclerosis Society and Peter Calabresi for the DTI tractography data.

References

Basser, P., Mattiello, J., and LeBihan, D. (1994). Mr diffusion tensor spectroscopy and imaging. *Biophysical Journal*.

Basser, P., Pajevic, S., Pierpaoli, C., and Duda, J. (2000). In vivo fiber tractography using dt-mri data. *Magnetic Resonance in Medicine*.

Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional Linear Model. *Statist. & Prob. Letters*, 45:11–22.

Cardot, H., Ferraty, F., and Sarda, P. (2003). Spline Estimators for the Functional Linear Model. *Statistica Sinica*.

Cardot, H. and Sarda, P. (2005). Estimation in Generalized Linear Model for Functional Data via Penalized Likelihood. *Journal of Multivariate Analysis*.

Crainiceanu, C. and Goldsmith, A. (2009). Bayesian Functional Data Analysis using WinBUGS. *Johns Hopkins University, Dept. of Biostatistics Working Papers*.

Crainiceanu, C., Staicu, A., and Di, C. (2008). Generalized Multilevel Functional Regression. *Johns Hopkins University, Dept. of Biostatistics Working Papers*, page 173.

Di, C., Crainiceanu, C., Caffo, B., and Punjabi, N. (2008). Multilevel Functional Principal Component Analysis.

Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer.

James, G. (2002). Generalized linear models with functional predictors. *Journal Of The Royal Statistical Society Series B*, 64(3):411–432.

LeBihan, D., Mangin, J., Poupon, C., and Clark, C. (2001). Diffusion tensor imaging: Concepts and applications. *Journal of Magnetic Resonance Imaging*.

Li, Y. and Ruppert, D. (2008). On the asymptotics of penalized splines. *Biometrika*, 60(95):415–436.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman & Hall/CRC.

Mori, S. and Barker, P. (1999). Diffusion magnetic resonance imaging: its principle and applications. *Anat Rec (New Anat)*.

Muller, H. and Stadtmuller, U. (2005). Generalized functional linear models. *Annals of Statistics*, 33(2):774–805.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems (with discussion). *Statistical Science*, (1):505–527.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer.

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

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11(4):23.

Ruppert, D., Wand, M., and Carroll, R. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

Staniswalis, J. and Lee, J. (1998). Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall.

Yao, F., Müller, H., Clifford, A., Dueker, S., Follett, J., Lin, Y., Buchholz, B., and Vogel, J. (2003). Shrinkage estimation for functional principal component scores with application to the population. *Biometrics*.