Published in final edited form as:
J Comput Graph Stat. 2011 December 1; 20(4): 830–851. doi:10.1198/jcgs.2010.10007.

Penalized Functional Regression

Jeff Goldsmith,
Johns Hopkins Bloomberg School of Public Health Biostatistics, Baltimore, MD 21205

Jennifer Bobb,
Johns Hopkins Bloomberg School of Public Health Biostatistics, Baltimore, MD 21205

Ciprian M. Crainiceanu,
Johns Hopkins Bloomberg School of Public Health Biostatistics, Baltimore, MD 21205

Brian Caffo, and
Johns Hopkins Bloomberg School of Public Health Biostatistics, Baltimore, MD 21205

Daniel Reich
Department of Radiology and Imaging Sciences, National Institutes of Health, Bethesda, Maryland 20892-1074

Jeff Goldsmith: jgoldsmi@jhsph.edu; Jennifer Bobb: jfeder@jhsph.edu; Ciprian M. Crainiceanu: ccrainic@jhsph.edu; Brian Caffo: bcaffo@jhsph.edu; Daniel Reich: daniel.reich@nih.gov

Abstract

We develop fast fitting methods for generalized functional linear models. The functional predictor is projected onto a large number of smooth eigenvectors and the coefficient function is estimated using penalized spline regression; confidence intervals based on the mixed model framework are obtained. 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. The approach can be implemented using standard mixed effects software and is computationally fast. The methodology is motivated by a study of white-matter demyelination via diffusion tensor imaging (DTI). 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 (James 2002; Cardot, Ferraty, and Sarda 2003; Cardot and Sarda 2005; James and Silverman 2005; Müller 2005; Ramsay and Silverman 2005; Ferraty and Vieu 2006; Reiss and Ogden 2007; Crainiceanu, Staicu, and Di 2009). In this article we develop a novel inferential approach to functional regression. Our goals are to: (1) reduce the number of tuning parameters used in functional regression; (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) in an associated mixed effects model. Methods are implemented using standard mixed effects software.

Several important advantages of our penalized functional regression (PFR) approach are that (1) it provides a unified framework for functional regression in many settings, including when functions are measured with error, at equal or unequal intervals, at a dense or sparse set of points, and to multiple functional regressors observed at one or multiple levels; (2) it is computationally efficient compared to other penalized approaches and can be fit using standard software; (3) it is automated in that the smoothing parameter is estimated as a variance component in a mixed effects model, avoiding manual selection or a cross-validation procedure; and (4) confidence intervals based on mixed effects inferential machinery are readily constructed. Moreover, our methods apply to outcomes distributed in the exponential family class of models.

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 on functional 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 (McCullagh and Nelder 1989; Cardot and Sarda 2005; Müller and Stadtmüller 2005)

$$\begin{aligned} Y_i &\sim \mathrm{EF}(\mu_i, \eta), \\ g(\mu_i) &= \alpha + \int_0^1 X_i(s)\beta(s)ds + \mathbf{Z}_i\gamma. \end{aligned} \tag{1.1}$$

Here $\mathrm{EF}(\mu_i, \eta)$ 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, $W_i(t) = X_i(t) + \varepsilon_i(t)$, where $\varepsilon_i \sim N[0, \sigma_\varepsilon^2]$. 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 (1.1) including the function $\beta(\cdot)$, which characterizes the relationship between the transformed mean of Y and the covariate of interest $X(\cdot)$. Because of some hesitation to adopt model (1.1), on the part of both scientific collaborators and statisticians, we feel it is worth explaining the interpretation of the integral appearing there. Consider a coarse partition $T = \{T_1, \ldots, T_G\}$ of the interval $[0, 1]$, and for the moment assume $\beta(t) = \beta_j$ for $t \in T_j$; that is, assume $\beta(t)$ is a step function on T. Then

$$\int_0^1 X_i(t)\beta(t)dt = \sum_{j=1}^{G} \beta_j \int_{t \in T_j} X_i(t)dt = \sum_{j=1}^{G} \beta_j \bar{X}_{ij},$$

where $\bar{X}_{ij}$ is the mean of $X_i(t)$ on T_j. An extreme example takes $\beta(t) = \beta$ if $t \leq 0.5$ and 0 otherwise. In this case $\int_0^1 X_i(t)\beta(t)dt = \beta \int_0^{0.5} X_i(t)dt$ and model (1.1) becomes a standard regression model which contains the average functional covariate over the interval [0, 0.5] as a regressor. Considered this way, $\beta(\cdot)$ provides weights that are applied to all subject level functions in the same manner. Intuition for the integral in model (1.1) is enhanced by contemplating finer and finer partitions T. In practice it makes sense to consider smooth transitions in the weighting scheme, that is, a smooth $\beta(\cdot)$ function. Thus, we think of $\beta(\cdot)$ as the smooth weighting scheme which, when applied to the subject-specific predictors $X_i(\cdot)$, is most predictive of the outcome. 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.

Our proposed approach to estimating the coefficient function $\beta(t)$ has two steps (the following uses notation from Ramsay and Silverman 2005, chap. 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 $\boldsymbol{\psi} = \{\psi_1(t), \ldots, \psi_{K_x}(t)\}$ is the collection of the first K_x eigenfunctions of the smoothed covariance matrix $\Sigma^X(s, t) = \text{cov}[X_i(s), X_i(t)]$. Second, we use a truncated power series spline basis $\boldsymbol{\varphi}(t) = \{\varphi_1(t), \ldots, \varphi_{K_b}(t)\}$ for $\beta(t)$, so that $\beta(t) = \boldsymbol{\varphi}(t)\mathbf{b}$. 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 functional regression literature, the choice of K_x is particularly important when using a low-dimensional approach that uses $\boldsymbol{\psi}$ for both the predictors and the coefficient: it must be large enough that $\beta(t)$ is in the space spanned by $\{\psi_1(t), \ldots, \psi_{K_x}(t)\}$ but small enough to smoothly estimate $\beta(t)$. In the smoothing literature, the choice of the number of knots in $\boldsymbol{\varphi}$ has been shown (Ruppert 2002; Li and Ruppert 2008) 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 explicit smoothness constraint 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 from the smoothing literature in the current functional setting and choose K_b large; K_x is chosen large enough to satisfy the identifiability constraint $K_x \geq K_b$. Choosing a large K_x avoids having to choose 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 (1.1) 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 by Cardot, Ferraty, and Sarda (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 $X_i(t)$. We also greatly improve the computation time associated with fitting a penalized model by taking advantage of the link to mixed effects models and existing well-tested software. The same ideas can be extended seamlessly to functional regression when the exposure proxy has a multilevel structure (Crainiceanu, Staicu, and Di 2009; Di et al. 2009). Thus, by decoupling the estimation of the functional predictors and the coefficient function, and by exploiting the connection to mixed models, our approach leads

to straightforward extensions and direct integration with other functional regression settings. Importantly, the connection to mixed models is further exploited to develop confidence intervals: while Müller and Stadtmüller (2005) derived confidence intervals for a low-dimension approach to functional regression and others (Reiss and Ogden 2007; James, Wang, and Zhu 2009) discussed empirical or bootstrap confidence intervals, we are unaware of existing explicit derivations of confidence intervals based on a penalized approach to functional regression.

It is important to distinguish our use of the PC decomposition from the widely used functional principal components regression (FPCR) techniques (Cardot, Ferraty, and Sarda 1999; Reiss and Ogden 2007). Stated shortly, FPCR regresses the vector of scalar outcomes Y on the design matrix $\mathbf{XV}_{K_X}$, where $\mathbf{X}$ has ith row $[X_i(t_1), \ldots, X_i(t_T)]$, $\mathbf{V}_{K_X}$ is the truncated at K_X version of the matrix $\mathbf{V}$ in $\mathbf{UDV}^T$, the singular value decomposition of $\mathbf{X}$. That is, FPCR regresses Y on the first K_X PC loadings of the functional regressors. Low-dimension FPCR techniques choose K_X either by hand or according to a rule; the resulting estimates can be very sensitive to this choice, and may be poor if $\beta(t)$ is not in the space spanned by the relatively few PCs. High-dimension penalized FPCR approaches often employ an explicit penalization constraint and choose K_X via cross-validation, which is computationally expensive and does not completely alleviate the concern that $\beta(t)$ is not in the space spanned by the basis. In our simulations we compare our method to both low-and high-dimensional FPCR approaches. 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 article is organized as follows. Section 2 provides the details of our approach to functional regression. Section 3 describes the seamless generalization to multiple functions, clustered functions, and to sparse functional data. Section 4 provides a detailed simulation to compare several methods in the univariate setting and explores the coverage probabilities of confidence intervals. We apply our method to the DTI data in Section 5, and conclude with a discussion in Section 6. Additional simulations for the multivariate, multilevel, and sparse functional data are found in an online appendix. 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 estimator 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 (1.1) using an expansion into the PC basis obtained from its covariance operator, $\Sigma^X(\cdot, \cdot)$. As mentioned in Section 1, choosing the number of components is often viewed as both important and difficult; here we elect to use a large number of PCs and largely avoid this issue. This refocuses the problem on estimating $\Sigma^X(\cdot, \cdot)$, which is a much simpler problem.

Assume that instead of observing $X_i(t)$ one measures a proxy $W_i(t) = X_i(t) + \varepsilon_i(t)$, where $\varepsilon_i \sim N[0, \sigma_\varepsilon^2]$. The covariance operator for the observed data is $\sum^W(s,t) = \sum^X(s,t) + \sigma_\varepsilon^2 \delta_{ts}$, where $\Sigma^W(s, t) = \text{cov}\{W_i(s), W_i(t)\}$ is the covariance operator on the observed functions, $\Sigma^X(s, t) = \text{cov}\{X_i(s), X_i(t)\}$, and $\delta_{ts} = 1$ if $t = s$ and is 0 otherwise. This suggests the following strategy for estimating $\Sigma^X(s, t)$. First, construct a method of moments estimator

$\hat{\Sigma}^W(s, t)$ of $\Sigma^W(s, t)$ from the observed data. Second, smooth $\hat{\Sigma}^W(s, t)$ 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 $\Sigma^W(t_{ik}, t_{il})$ 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 by Di et al. (2009) to estimate $\Sigma^W(s, t)$:

1. Use a very small bandwidth smoother to obtain a rough estimate of the covariance operator.
2. Use a fast automatic nonparametric smoother of the undersmoothed surface obtained at the previous step.

Let $\sum_{k=1}^{\infty} \lambda_k \psi_k(s)\psi_k(t)$ be the spectral decomposition of $\hat{\Sigma}^X(s, t)$, where $\lambda_1 \geq \lambda_2 \geq \cdots$ are the nonincreasing 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–Loéve 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 Müller and Stadtmüller (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, Staicu, and Di 2009; Di et al. 2009)

$$\begin{aligned} W_i(t_{ij}) &= \sum_{k=1}^{K_x} c_{ik}\psi_k(t_{ij}) + \varepsilon_{ij}, \\ c_{ik} \sim N(0, \sigma_c^2), &\quad \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2), \end{aligned} \tag{2.1}$$

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 by Crainiceanu, Staicu, and Di (2009). When this problem is a real concern a solution is to jointly model the outcome model and model (2.1). This approach can be addressed using a fully Bayesian analysis (Crainiceanu and Goldsmith 2010; Goldsmith et al. 2010) and is not the focus of this article. 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 on the problem of estimating $\beta(t)$ using a method that does not depend on the particular choice of number of principal components, a nontrivial distinction.

2.2 Estimation of $\boldsymbol{\beta(t)}$

The second step in our method is modeling $\beta(t)$ and we borrow ideas from the penalized spline literature (O'Sullivan, Yandell, and Raynor 1986; Ruppert, Wand, and Carroll 2003;

Wood 2006). Let $\boldsymbol{\varphi}(t) = \{\varphi_1(t), \varphi_2(t), \ldots, \varphi_{K_b}(t)\}$ be a spline basis, so that $\beta(t)=\sum_{k=1}^{K_b} b_k\varphi_k(t)=\boldsymbol{\varphi}(t)\mathbf{b}$, where $\mathbf{b} = \{b_1, \ldots, b_{K_b}\}^T$. Thus, the integral in model (1.1) becomes

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

where $\mathbf{c}_i'=(c_{i1},\ldots,c_{iK_x})$ and $\mathbf{J}_{\psi\varphi}$ is a $K_x \times K_b$ dimensional matrix with the (k, l)th entry equal to $\int_0^1 \psi_k(s)\varphi_l(s)\,ds$ (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; (2) the smoothing of the $\beta(\cdot)$ function is implicitly controlled by K_x, the smoothing parameter for the functional process, $X_i(t)$, and can be very sensitive to the choice of K_x; and (3) the choice of K_x is typically undertaken either by hand, which can be subjective, or via cross-validation, which is computationally expensive. Thus, we use the truncated power series spline basis expansion $\beta(t)=b_0+b_1t+\sum_{k=3}^{K_b} b_k(t-\kappa_k)_+$ where $\{\kappa_k\}_{k=3}^{K_b}$ are knots. We explicitly induce smoothing by assuming that $\{b_k\}_{k=3}^{K_b} \sim N(0, \sigma_b^2\mathbf{I})$. Other choices of basis functions can be used with corresponding changes to the penalty matrix.

Denote by $\mathbf{C}$ the $I \times K_x$ dimensional matrix of PC loadings with 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 (1.1) can be reformulated in matrix format as

$$\begin{gathered}\boldsymbol{Y}|\mathbf{X}(\mathbf{t}) \sim \mathrm{EF}(\mu,\gamma),\\ g(\mu)=[\;1\quad \mathbf{C}\mathbf{J}_{\psi\varphi}\quad \boldsymbol{Z}\;][\;\alpha\quad \mathbf{b}\quad \gamma\;]^T,\\ \{b_k\}_{k=3}^{K_b} \sim N[0,\sigma_b^2\mathbf{I}],\end{gathered} \tag{2.2}$$

which is a mixed effects model with $K_b - 2$ random effects, $\{b_k\}_{k=3}^{K_b}$. This model can be fit robustly using standard mixed effects software (Ruppert 2002; McCulloch, Searle, and Neuhaus 2008).

Intuition for the connection between the mixed model representation of the functional regression model and the induced smoothness of $\hat{\beta}(t)$ can be built most easily when the outcome Y_i is normally distributed. In this case, maximization of the likelihood of $(\mathbf{Y}, \mathbf{b})$ over the unknowns $(\alpha, \mathbf{b}, \boldsymbol{\gamma})$ is equivalent to minimizing

$$\frac{1}{\sigma_y^2}(y-\alpha-CJ\mathbf{b}-Z\gamma)^T(y-\alpha-CJ\mathbf{b}-Z\gamma)+\frac{1}{\sigma_b^2}\mathbf{b}^T D\mathbf{b},$$
$$\text{where } D=\begin{bmatrix}\mathbf{0}_{2\times 2} & \mathbf{0}_{2\times K_b}\\ \mathbf{0}_{K_b\times 2} & \mathbf{I}_{K_b\times K_b}\end{bmatrix}.$$

This expression contains an explicit penalty on the spline terms $\{b_k\}_{k=3}^{K_b}$. Moreover, the mixed model framework allows us to write the joint likelihood of all model parameters (including

the smoothing parameter σ_b^2) and to estimate these parameters using maximum likelihood and restricted maximum likelihood techniques.

Once the basis for $\beta(t)$ is chosen, model (2.2) depends on the choice of K_b and K_x. Following Ruppert (2002) we select K_b large enough to prevent undersmoothing and, to satisfy the identifiability constraint, select $K_x \geq K_b$. A rule of thumb is to choose $K_b = K_x = 35$; however, the specific value of K_b is unimportant as long as it is large enough to capture the maximum variability in $\beta(t)$. The position of the knots in the truncated power series spline basis is typically unimportant and we place them at the quantiles of the distribution of t_{ij}.

It is important to note that the smoothing parameter σ_b^2 is estimated as a variance component in the mixed effects model. Therefore, the only tuning parameters chosen by the user are K_b, K_x; as long as these are chosen large enough, their specific value has little impact on estimation. Thus, the procedure is highly automated and robust to changes in the selection of the tuning parameters.

We also point out that complexity of fitting model (2.2) 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). In our simulations, we use the nlme package in R to fit model (2.2); this package uses first a moderate number of EM iterations to refine starting values of the variance components followed by a Newton–Raphson optimization of the restricted log-likelihood (Bates and Pinheiro 1998; Pinheiro et al. 2009; R Development Core Team 2009). Because our approach takes advantage of established mixed model theory and software, it is computationally efficient compared to penalized approaches to functional regression that employ a cross-validation step to choose the smoothing parameter.

2.3 Confidence Intervals for $\beta(t)$

Because model (2.2) 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, Wand, and Carroll 2003; Wood 2006).

For illustration, consider the case when $Y_i = \alpha + \int_0^T X_i(t)\beta(t)dt + \varepsilon_i$ with $\varepsilon_i \sim N[0, \sigma_\varepsilon^2]$. Take as the basis for $\beta(t)$ the functions 1, t, $(t - \kappa_2)_+$, …, $(t - \kappa_{K_b})_+$. Let $\boldsymbol{\beta} = [\alpha\ \mathbf{b}^T]^T$; it is easy to show that $\hat{\boldsymbol{\beta}} = (\mathbf{W}^T\mathbf{R}^{-1}\mathbf{W} + \mathbf{B})^{-1}\mathbf{W}^T\mathbf{Y}$, where $\mathbf{W} = [1\ \mathbf{C}\mathbf{J}_{\psi\varphi}]$,

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times K} \\ \mathbf{0}_{K\times 2} & \mathbf{G}^{-1} \end{bmatrix},$$

$\mathbf{R} = \sigma_\varepsilon^2\mathbf{I}_n$, and $\mathbf{G} = \sigma_b^2\mathbf{I}_{K_b-2}$. The variance components σ_ε^2, σ_b^2 can be estimated via REML in the corresponding mixed effects model (2.2). Recall that $\beta(t) = \boldsymbol{\varphi}(t)\mathbf{b}^T$. One can show that

$$\text{var}[\widehat{\beta} - \beta] = (\mathbf{W}^T\mathbf{R}^{-1}\mathbf{W} + \mathbf{B})^{-1} = \begin{bmatrix} \sum_{\alpha\alpha} & \sum_{\mathbf{b}\alpha} \\ \sum_{\mathbf{b}\alpha}^T & \sum_{\mathbf{bb}} \end{bmatrix}.$$

Then, at any t_0 $\text{var}[\hat{\beta}(t_0)] = \text{var}[\boldsymbol{\varphi}(t_0)\hat{\mathbf{b}}^T] = \boldsymbol{\varphi}(t_0)\Sigma_{\mathbf{bb}}\boldsymbol{\varphi}(t_0)^T$; we estimate $\widehat{\text{sd}}\{\widehat{\beta}(t_0)\} = \sqrt{\boldsymbol{\varphi}(t_0)\widehat{\sum}_{\mathbf{bb}}\boldsymbol{\varphi}(t_0)^T}$, where $\hat{\boldsymbol{\Sigma}}_{\mathbf{bb}}$ is the $(K_b) \times (K_b)$ dimensional matrix obtained by plugging the REML estimates for the variance components into the formula for $\text{var}[\hat{\boldsymbol{\beta}}]$. An approximate 95% confidence interval for $\text{E}[\hat{\beta}(t_0)]$ can be constructed as $\widehat{\beta}(t_0) \pm 1.96\widehat{\text{sd}}\{\widehat{\beta}(t_0)\}$.

We point out some limitations of this derivation. The first is that the confidence intervals will perform poorly in regions where $\hat{\beta}(t)$ oversmooths the true coefficient function and is therefore biased; a possible solution would be an adaptive smoothing approach, which we do not consider here. A second limitation is that we neglect the variability in estimating the PC loadings in **C**; in many applications, this variability is minimal but, as pointed out by a reviewer, could be substantial when the regressors are observed sparsely at the subject level or with error. One solution would consider modeling the PC loadings and the coefficient function jointly, as in the works of Crainiceanu and Goldsmith (2010) and Goldsmith et al. (2010), and using instead the 95% posterior credible interval for $\beta(t)$.

3. EXTENSIONS

As discussed earlier, a strength of the proposed method is the wide applicability that stems from treating the estimation of $\beta(t)$ and the $X_i(t)$ separately. In this section, we demonstrate this flexibility by exploring extensions to several common functional regression settings.

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 (1.1) is given by

$$\begin{aligned} Y_i &\sim \text{EF}(\mu_i, \eta), \\ g(\mu_i) &= \alpha + \int_0^1 X_{i1}(s)\beta_1(s)ds + \cdots + \int_0^1 X_{iL}(s)\beta_L(s)ds + \mathbf{Z}_i\gamma. \end{aligned} \tag{3.1}$$

The approach given in Section 2 extends naturally to the multivariate functional regression setting. That is, we once again treat the estimation of the $X_{il}(t)$ and of the $\beta_l(t)$ as separate. First, approximate each functional regressor using the truncated Karhunen–Loéve decomposition, $X_{il}(t) = \sum_{k=1}^{K_x} c_{ikl}\psi_{kl}(t)$, where $\boldsymbol{\psi}_l(\cdot) = \{\psi_{kl}(\cdot): 1 \le k \le K_x\}$ are the first eigenfunctions of the smoothed estimated covariance operator $\sum_l^X(s, t) = \text{cov}\{X_{il}(s), X_{il}(t)\}$. Then, express each coefficient function in model (3.1) in terms of a spline basis $\boldsymbol{\varphi}(t) = \{\varphi_1(t), \varphi_2(t), \ldots, \varphi_{K_b}(t)\}$, so that $\beta_l(t) = \sum_{k=1}^{K_b} b_{lk}\varphi_k(t)$. We therefore have that

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

where $\mathbf{c}_{il}' = (c_{i1l}, \ldots, c_{iK_xl})^T$ and $\mathbf{J}_l$ is a $K_x \times K_b$ dimensional matrix with the (k, m)th entry equal to $\int_0^1 \psi_{kl}(t)\varphi_m(t)dt$. As in Section 2.2, we assume the use of a truncated power series

spline basis for each $\beta_l(t)$ and induce smoothness on the estimate of $\beta_l(t)$ by assuming $\{b_{lk}\}_{k=3}^{K_b} \sim N[0, \sigma_{b_l}^2 \mathbf{I}]$.

The multivariate functional regression model can be expressed as the mixed effects model

$$\begin{aligned} \boldsymbol{Y}|\mathbf{X}(\mathbf{t}) &\sim \mathrm{EF}(\mu, \gamma), \\ g(\mu) = \begin{bmatrix} 1 & \mathbf{C}_1\mathbf{J}_1 & \cdots & \mathbf{C}_L\mathbf{J}_L & \boldsymbol{Z} \end{bmatrix} \begin{bmatrix} \alpha & \mathbf{b}_1 & \cdots & \mathbf{b}_L & \gamma \end{bmatrix}^T, \\ \{b_{lk}\}_{k=3}^{K_b} &\sim N[0, \sigma_{b_l}^2 \mathbf{I}], \quad l = 1, \ldots, L, \end{aligned} \tag{3.2}$$

with $K_b - 2$ random effects, $\{b_{lk}\}_{k=3}^{K_b}$, for each functional coefficient $\beta_l(t)$and can be fit using standard mixed models 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 $\varphi_l(t) = \{\varphi_{1l}(t), \varphi_{2l}(t), \ldots, \varphi_{K_b l}(t)\}$ as the basis for $\beta_l(t)$, the matrix J_l has (k, m)th entry equal to $\int_0^1 \psi_{kl}(t)\varphi_{ml}(t)dt$; 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 the work of Crainiceanu, Staicu, and Di (2009). This extension is informative in that although the estimation of the (unobserved) $X_i(t)$ is difficult, the same framework that we have developed applies unchanged here. That is, first we estimate the $X_i(t)$ based on a functional principal components decomposition, then we estimate $\beta(t)$ using a rich spline basis and an explicit smoothing parameter controlled via a mixed model.

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 $j = 1, 2, \ldots, J_i$. We assume that $W_{ij}(t)$ is a proxy for the true underlying subject-specific function $X_i(t)$, so that $W_{ij}(t) = \mu(t) + \eta_j(t) + X_i(t) + U_{ij}(t) + \varepsilon_{ij}(t)$. 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 $\varepsilon_{ij} \sim N[0, \sigma_\varepsilon^2]$. We further assume that $X_i(t)$, $U_{ij}(t)$, and $\varepsilon_{ij}(t)$ are uncorrelated to guarantee identifiability. We construct $\hat{\mu}(t) = \bar{W}_{\cdot\cdot}(t)$ and $\hat{\eta}_j(t) = \bar{W}_{\cdot\cdot}(t) - \bar{W}_{\cdot j}(t)$, 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 $W_{ij}(t) = X_i(t) + U_{ij}(t) + \varepsilon_{ij}(t)$.

We use model (1.1) 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, Staicu, and Di 2009; Di et al. 2009) to construct parsimonious bases for $X_i(t)$, $U_{ij}(t)$. Following these articles, we first note that under certain assumptions the covariance operators $\Sigma^X = \mathrm{cov}[X_i(s), X_i(t)]$ and $\Sigma^U = \mathrm{cov}[U_{ij}(s), U_{ij}(t)]$ are given by $\mathrm{cov}[W_{ij}(s, t), W_{ik}$

$(s, t)]$ and $\text{cov}[W_{ij}(s, t), W_{ij}(s, t)] - \text{cov}[W_{ij}(s, t), W_{ik}(s, t)]$, respectively. Next, we calculate the spectral decompositions $\sum^{X}(s,t)=\sum_{k=1}^{\infty}\lambda_k^{(1)}\psi_k^{(1)}(s)\psi_k^{(1)}(t)$ and $\sum^{U}(s,t)=\sum_{l=1}^{\infty}\lambda_l^{(2)}\psi_1^{(2)}(s)\psi_l^{(2)}(t)$, where $\lambda_1^{(1)} \geq \lambda_2^{(1)} \geq \lambda_3^{(1)} \cdots$ and $\lambda_1^{(2)} \geq \lambda_2^{(2)} \geq \lambda_3^{(2)} \cdots$ 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 the article by Crainiceanu, Staicu, and Di (2009), we estimate c_{ik}, ζ_{ijk} using the mixed model

$$W_{ij}(t)=\sum_{k=1}^{K_x}c_{ik}\psi_j^{(1)}(t)+\sum_{l=1}^{L}\zeta_{ijl}\psi_l^{(2)}(t)+\varepsilon_{ij}(t), \tag{3.3}$$

$$c_{ik} \sim N[0, \lambda_k^{(1)}]; \quad \zeta_{ijl} \sim N[0, \lambda_l^{(2)}]; \quad \varepsilon_{ij} \sim N[0, \sigma_\varepsilon^2]. \tag{3.4}$$

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

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

Thus, the outcome model is identical to model (2.2), 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 models software is used.

This development is related to the one proposed by Crainiceanu, Staicu, and Di (2009). Specifically, the method of Crainiceanu, Staicu, and Di (2009) 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 in both standard simulation settings and applications.

3.3 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 $\cup_{i=1}^{I}[\{t_{ij}\}_{j=1}^{J_i}]$ is dense in $[0, 1]$. Here again, the $W_i(t)$ are measured-with-error proxies for the true $X_i(t)$ so that $W_i(t) = X_i(t) + \varepsilon_i(t)$, where $\varepsilon_i(t) \sim N(0, \sigma_\varepsilon^2)$. We again point out that while some care must be taken in the estimation of the $X_i(t)$, the same general procedure that has been used elsewhere applies here.

We use the following method, adapted from the work of Di et al. (2009), to estimate the subject-specific functional regressors based on a PC decomposition of the covariance operator $\Sigma^X(s, t)$. 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 $t_1, \ldots, t_S$, and for each subject let $t_{ij\,s}$ be the point in this grid nearest to the observed point t_{ij}. The covariance operator can be roughly estimated using $\widehat{\sum}_1^W(r, s) = \sum_{i \in I(r,s)} \{W_i(t_{ijr}) - \overline{X}(t_r)\}\{W_i(t_{ijs}) - \overline{X}(t_s)\}/N(r, s)$, where $\bar{X}(t)$ 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 rough covariance matrix to obtain $\hat{\Sigma}^X(r, s)$, the estimated covariance operator of the sparsely observed subject-specific functional regressors on our newly defined grid $t_1, \ldots, t_S$.

As before, the function $X_i(t)$ is approximated by $X_i(t) = \sum_{k=1}^{K_x} c_{ik}\psi_k(t)$, where K_x is the truncation lag, $\boldsymbol{\psi}(\cdot) = \{\psi_k(\cdot): 1 \le k \le K_x\}$ are the first eigenfunctions of the smoothed estimated covariance operator, 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:

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

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.

Using the same notation as in other settings, the integral in the linear predictor of model (1.1) has the matrix representation $\int_0^T X_i(t)\beta(t)dt = \mathbf{c}_i'\mathbf{J}_{\psi\varphi}\mathbf{b}$. 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.

4. SIMULATION

In this section, we pursue a simulation study to explore the viability of our method in the univariate functional regression setting. We compare our approach to several others, including functional principal components regression and penalized approaches, under several true coefficient functions and with varying levels of measurement error. We also include brief results regarding confidence interval performance. Additional simulations, in the multivariate, multilevel, and sparsely observed settings, are available in the online supplementary materials; the results are very similar to those in the univariate case. All code used to conduct simulations is also available online.

4.1 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:

$$\begin{aligned} Y_i &= \frac{1}{G}\sum_{g=1}^{G} X_i(t_g)\beta(t_g) + \varepsilon_i, \quad i = 1, \ldots, 200, \\ W_i(t_g) &= X_i(t_g) + \delta_i(t_g), \\ X_i(t_g) &= u_{i1} + u_{i2}t_g + \sum_{k=1}^{10}\left\{v_{ik1}\sin\left(\frac{2\pi k}{10}t_g\right) + v_{ik2}\cos\left(\frac{2\pi k}{10}t_g\right)\right\}, \end{aligned} \tag{4.1}$$

where $\varepsilon_i \sim N[0, \sigma_\varepsilon^2]$, $\delta_i(t_g) \sim N[0, \sigma_x^2]$, $u_{i1} \sim N[0, 25]$, $u_{i2} \sim N[0, 0.04]$, and $v_{ik1}, v_{ik2} \sim N[0, 1/k^2]$. For reference, Figure 1 displays a sample of 10 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 the work of Müller and Stadtmüller (2005). The first few PCs 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, we consider $\sigma_\varepsilon^2 \in \{0.5, 1\}$ and three true coefficient functions $\beta(\cdot)$, yielding 12 possible parameter combinations. The choices of the coefficient functions $\beta(\cdot)$ are described below.

For each combination of the parameter values $\sigma_\varepsilon^2, \sigma_x^2$, and $\beta(\cdot)$, we simulate 1000 datasets $[Y_i, W_i(t_g) : i = 1, \ldots, 200]$. We compare four alternative approaches to estimating $\beta(\cdot)$ to our approach as described in Section 2. Performance in estimating $\beta(\cdot)$ is compared by calculating the average mean squared error (AMSE) over the 1000 samples as

$$\mathrm{AMSE}(\widehat{\beta}(\cdot)) = \frac{1}{1000}\sum_{r=1}^{1000}\left[\frac{1}{G}\sum_{g=0}^{G}\{\widehat{\beta}_r(t_g) - \beta(t_g)\}^2\right],$$

where $\hat{\beta}_r(\cdot)$ is the estimated coefficient function from the rth simulated dataset.

The first method for estimating $\beta(\cdot)$ is principal components regression (PCR). Let $\mathbf{X}$ be the $200 \times G$ matrix with ith row $(X_i(t_1), \ldots, X_i(t_G))$ and calculate the singular value decomposition $\mathbf{UDV}^T$ of $\mathbf{X}$. In PCR, the scalar outcomes Y are regressed on $\mathbf{V}_A$, the $200 \times A$ matrix containing the first A columns of $\mathbf{V}$, that is, 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 $\sum_{i=1}^{n}(Y_i - \widehat{Y}_{-i})^2$ is minimized. Here $\hat{Y}_{-i}$ is the predicted value for the ith data point obtained from fitting the PCR model to the data with the ith observation deleted. Though this procedure is computationally intensive, we implement a faster alternative formulation for the statistic for a linear model, namely,

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

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

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

where λ_a is the eigenvalue corresponding to the ath 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 $X_i(t)$. Further, in the presence of measurement error, we use smoothed principal components.

The third 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 $\mathbf{XB}$. A penalized regression model is then fit to find ξ that minimizes the criterion

$$\left\| Y - \mathbf{XBV}_A \xi \right\|^2 + \lambda \xi^T \mathbf{V}_A^T P^T P \mathbf{V}_A \xi,$$

where $\mathbf{V}_A$ is the first A columns of $\mathbf{V}$ from the singular value decomposition $\mathbf{XB} = \mathbf{UDV}^T$ and P is the penalization matrix such that $\xi^T \mathbf{V}_A^T P^T P \mathbf{V}_A \xi$ approximately equals the integrated squared second derivative of the coefficient function $\xi^T \mathbf{V}_A^T$. 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 are 1–10, 12, and 15–40 at intervals of 5.

Finally, we implement the method SPCR-GCV (Cardot, Ferraty, and Sarda 2003), using code provided by the authors. This approach first computes the PCR estimate using the first K principal components of the $X_i(t)$ and then smooths 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 $\beta_1(t) = \sin(\pi t/5)$, $\beta_2(t) = (t/2.5)^2$, and $\beta_3(t) = -p(t \mid 2, 0.3) + 3p(t \mid 5, 0.4) + p(t \mid 7.5, 0.5)$, where $p(\cdot \mid \mu, \sigma)$ 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.

Table 1 compares the AMSE for each set of the parameters across approaches. 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 than both the FPCR$_R$ and PCR-PVE methods when $\sigma_x^2=0$ and performs comparably well or slightly better when $\sigma_x^2=1$; our method also has less than half the AMSE for β_1 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 outperformed SPCR-GCV which, in turn, performed much better than the other approaches. As expected, none of the methods performed particularly well for the third true coefficient function β_3; the FPCR$_R$ and SPCR-GCV methods provide the closest estimates, with our method performing slightly worse. In Figure 2 we plot the estimated beta functions from each approach that have the median AMSE for the case where $\sigma_x^2=0$ and $\sigma_\varepsilon^2=1$. 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 examine the case where $\sigma_x^2=0$ and $\sigma_\varepsilon^2=1$. 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 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 snowball in more complex settings. In fact, we are unaware of competing penalized approaches that have been generalized to each of the settings considered in this article. The computational issues pointed out above are a possible reason for this.

4.1.1 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 1000 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 $\sigma_\varepsilon^2=0.5$ with and without measurement error on the $X_i(t)$; taking $\sigma_\varepsilon^2=1$ yields similar confidence interval coverage probabilities.

These simulations illustrate both the strengths and limitations of the confidence interval procedure. For $\beta_1(t)$ and $\beta_2(t)$, we see generally good coverage when there is no measurement error. In fact, in some situations the intervals are slightly conservative. However, for $\beta_2(t)$ and to some degree $\beta_1(t)$ the performance degrades in the presence of measurement error; we speculate that this is in part due to the treatment of **C** as fixed. Thus, the simulations suggest that the confidence interval procedure is useful in situations with little measurement error on the functional predictor, but may be unreliable with larger measurement error variance. For $\beta_3(t)$, the coverage is expectedly poor: the estimate $\hat{\beta}(t)$

significantly oversmooths the true coefficient function, and confidence intervals based on this estimate fail to achieve nominal coverage rates.

5. 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 an 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, Mattiello, and LeBihan 1994Basser, Mattiello, and LeBihan 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. The top-left panel of Figure 4 shows the parallel diffusivity profile of a single tract for three cases and three controls.

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 4 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; also included in this plot are the leave-one-out cross-validated ROC curves for both the PFR and bin-mean methods. Note that there is a much larger drop in performance in the cross-validated curves for the bin-mean method than for PFR, perhaps indicating that the bin-mean approach is less generalizable to new datasets.

For each subject, we also compute the linear predictor $\int X_i(t)\beta(t)\,dt$ from the PFR method; the middle-right panel of Figure 4 shows the distribution of these quantities for 3 both cases and controls. As anticipated, $\int X_i(t)\beta(t)\,dt$ provides a reasonable quantity for distinguishing cases from controls based on the tract profile. The far-right panel of Figure 4 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 and 40 and above average between distances 50 and 65, and conversely for profiles with high linear predictors.

6. DISCUSSION

By combining several well-known techniques in functional data analysis, we have developed a method for generalized functional regression with the following properties: (i) flexibly estimates $\beta(t)$; (ii) treats the settings of measurement error, multilevel observations, and sparse data from the same unified framework; and (iii) compares favorably with existing methods in simulation studies. Although it builds on existing work, this method is conceptually new in that we decouple the estimation of the regressor functions $X_i(t)$ and the coefficient function $\beta(t)$. By using a PC decomposition for the $X_i(t)$, we are able to apply our method when the $X_i(t)$ are poorly observed (measurement error, sparse observation) or unobserved (multilevel). By using a rich spline basis for $\beta(t)$ and explicitly inducing smoothness, we are able to estimate arbitrary smooth coefficient functions. 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 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, appearing in the online appendix, 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 biased estimates of the predictor processes $X_t(t)$ and, ultimately, poor estimates of $\beta(t)$; however, fixing $\sigma_\varepsilon^2 = 0.05$ 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 datasets 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. A more robust development of confidence intervals for functional coefficients is also needed. Finally, the exploration methods in which the outcome is functional will continue to be important.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

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. Bobb's research was supported in part by training grant 2T32ES012871, from the U.S. NIH National Institute of Environmental Health Sciences. 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, LeBihan D. MR Diffusion Tensor Spectroscopy and Imaging. Biophysical Journal. 1994; 66:259–267. [PubMed: 8130344]

Basser P, Pajevic S, Pierpaoli C, Duda J. In vivo Fiber Tractography Using DT-MRI Data. Magnetic Resonance in Medicine. 2000; 44:625–632. [PubMed: 11025519]

Bates D, Pinheiro J. Computational Methods for Multilevel Modelling. 1998

Cardot H, Sarda P. Estimation in Generalized Linear Model for Functional Data via Penalized Likelihood. Journal of Multivariate Analysis. 2005; 92:24–41.

Cardot H, Ferraty F, Sarda P. Functional Linear Model. Statistics and Probability Letters. 1999; 45:11–22.

Cardot H, Ferraty F, Sarda P. Spline Estimators for the Functional Linear Model. Statistica Sinica. 2003; 13:571–591.

Crainiceanu C, Goldsmith J. Bayesian Functional Data Analysis Using WinBUGS. Journal of Statistical Software. 2010; 32:1–33.

Crainiceanu C, Staicu A, Di C. Generalized Multilevel Functional Regression. Journal of the American Statistical Association. 2009; 104:1550–1561. [PubMed: 20625442]

Di C, Crainiceanu C, Caffo B, Punjabi N. Multilevel Functional Principal Component Analysis. The Annals of Applied Statistics. 2009; 4:458–288. [PubMed: 20221415]

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

Goldsmith J, Crainiceanu C, Caffo B, Reich D. Longitudinal Penalized Functional Regression. 2010 to appear.

James G. Generalized Linear Models With Functional Predictors. Journal of the Royal Statistical Society, Ser B. 2002; 64:411–432.

James G, Silverman B. Functional Adaptive Model Estimation. Journal of the American Statistical Association. 2005; 100:565–576.

James G, Wang J, Zhu J. Functional Linear Regression That's Interpretable. The Annals of Statistics. 2009; 37:2083–2108.

LeBihan D, Mangin J, Poupon C, Clark C. Diffusion Tensor Imaging: Concepts and Applications. Journal of Magnetic Resonance Imaging. 2001; 13:534–546. [PubMed: 11276097]

Li Y, Ruppert D. On The Asymptotics Of Penalized Splines. Biometrika. 2008; 95:415–436.

McCullagh, P.; Nelder, J. Generalized Linear Models. Chapman & Hall/CRC; 1989.

McCulloch, C.; Searle, S.; Neuhaus, J. Generalized, Linear, and Mixed Models. Wiley; 2008.

Mori S, Barker P. Diffusion Magnetic Resonance Imaging: Its Principle and Applications. The Anatomical Record. 1999; 257:102–109. [PubMed: 10397783]

Müller H. Functional Modelling and Classification of Longitudinal Data. Scandinavian Journal of Statistics. 2005; 32:223–240.

Müller H, Stadtmüller U. Generalized Functional Linear Models. The Annals of Statistics. 2005; 33:774–805.

O'Sullivan F, Yandell B, Raynor W. Automatic Smoothing of Regression Functions in Generalized Linear Models. Journal of the American Statistical Association. 1986; 81:96–103.

Pinheiro, J.; Bates, D.; DebRoy, S.; Sarkar, D. the R Core Team. R package version 3.1–96. 2009. nlme: Linear and Nonlinear Mixed Effects Models.

R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing; Vienna, Austria: 2009. available at http://www.R-project.org. [8]

Ramsay, J.; Silverman, B. Functional Data Analysis. Springer; 2005.

Reiss P, Ogden R. Functional Principal Component Regression and Functional Partial Least Squares. Journal of the American Statistical Association. 2007; 102:984–996.

Ruppert D. Selecting the Number of Knots for Penalized Splines. Journal of Computational and Graphical Statistics. 2002; 11:735–757.

Ruppert, D.; Wand, M.; Carroll, R. Semiparametric Regression. Vol. 66. Cambridge University Press; 2003.

Staniswalis J, Lee J. Nonparametric Regression Analysis of Longitudinal Data. Journal of the American Statistical Association. 1998; 444:1403–1418.

Wood, S. Generalized Additive Models: An Introduction With R. Chapman & Hall; 2006.

Yao F, Müller H, Clifford A, Dueker S, Follett J, Lin Y, Buchholz B, Vogel J. Shrinkage Estimation for Functional Principal Component Scores With Application to the Population. Biometrics. 2003; 59:676–685. [PubMed: 14601769]

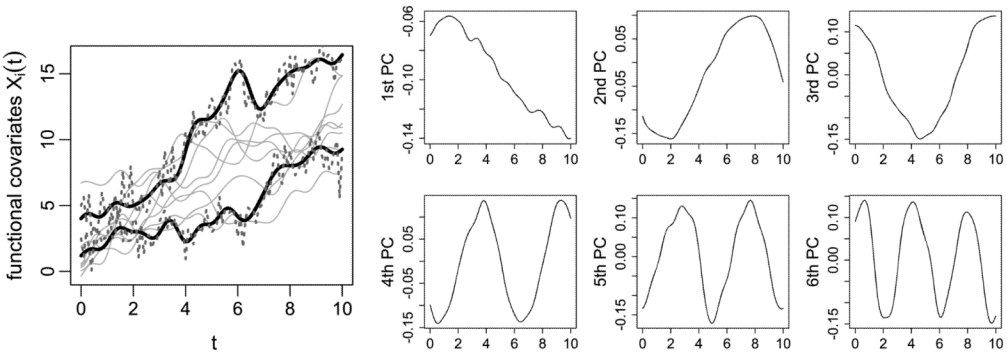

Figure 1.

The left panel displays a sample of 10 random functions generated from (4.1), 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 six estimated principal components. The online version of this figure is in color.

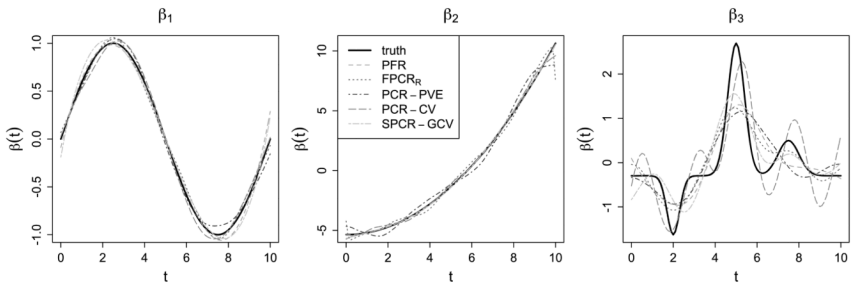

Figure 2.

For the simulation with $\sigma_X^2=0$ and $\sigma_\varepsilon^2=1$, we plot the estimated beta functions $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ from each method that have the median AMSE. The online version of this figure is in color.

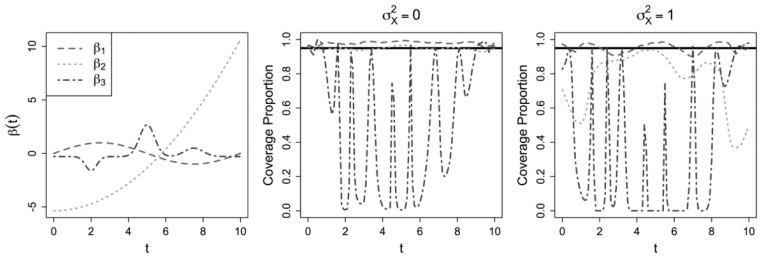

Figure 3.

For the simulations with σ_ε^2=0.5, a plot of the pointwise coverage probabilities for each true $\beta(t)$, shown without and with measurement error in the middle and right panels, respectively. The online version of this figure is in color.

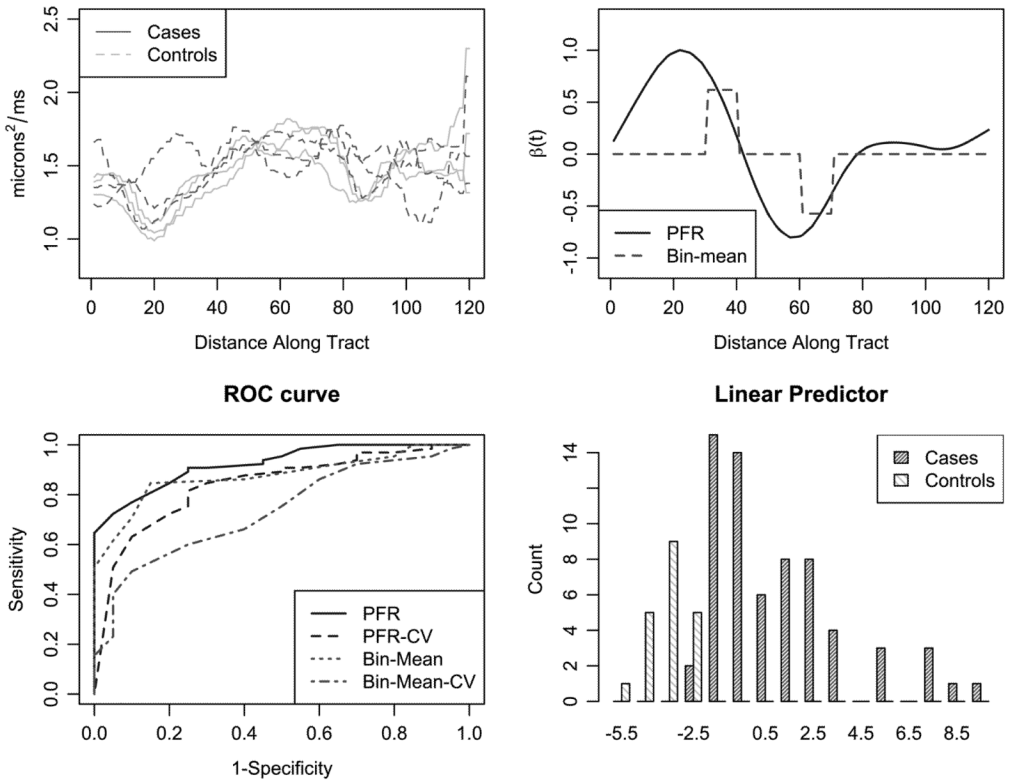

Figure 4.

The top-left panel shows a sample of functional predictors separated by case status. The top-right panel shows the estimated $\beta(t)$ from the PFR and bin-mean approaches. The bottom-left panel shows the ROC curves generated by these approaches as well as leave-one-out cross-validated ROC curves. The bottom-right panel shows the distribution of the linear predictor $\int_0^1 \beta(t)X_i(t)dt$ for the PFR method. The online version of this figure is in color.

Table 1

Average MSE over 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.

	$\beta_1(\cdot)$		$\beta_2(\cdot)$		$\beta_3(\cdot)$	
Method	σ_ε^2=0.5	σ_ε^2=1	σ_ε^2=0.5	σ_ε^2=1	σ_ε^2=0.5	σ_ε^2=1
PFR						
σ_X^2=0	0.0023	0.0037	2e–04	5e–04	0.19	0.234
σ_X^2=1	0.0034	0.0044	0.0607	0.0607	0.27	0.282
FPCRR						
σ_X^2=0	0.002	0.0025	0.207	0.239	0.15	0.194
σ_X^2=1	0.0054	0.0064	0.715	0.716	0.254	0.264
SPCR-GCV						
σ_X^2=0	0.0049	0.0084	0.0071	0.0114	0.158	0.178
σ_X^2=1	0.0076	0.0103	0.171	0.173	0.245	0.256
PCR-PVE						
σ_X^2=0	0.002	0.0032	0.264	0.266	0.289	0.29
σ_X^2=1	0.0035	0.0045	0.384	0.385	0.302	0.303
PCR-CV						
σ_X^2=0	0.0613	0.114	0.138	0.235	0.331	0.52
σ_X^2=1	0.0105	0.0136	0.479	0.466	0.258	0.27

Table 2

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

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