Joint Estimation of Monotone Curves via Functional Principal Component Analysis

Yei Eun Shin[a], Lan Zhou[b,*], Yu Ding[c]

[a] *Biostatistics Branch, Division of Cancer Epidemiology and Genetics, National Cancer Institute, USA*
[b] *Department of Statistics, Texas A&M University, USA*
[c] *Department of Industrial and Systems Engineering, Texas A&M University, USA*

Abstract

A functional data approach is developed to jointly estimate a collection of monotone curves that are irregularly and possibly sparsely observed with noise. In this approach, the unconstrained relative curvature curves instead of the monotone-constrained functions are directly modeled. Functional principal components are used to describe the major modes of variations of curves and allow borrowing strength across curves for improved estimation. A two-step approach and an integrated approach are considered for model fitting. The simulation study shows that the integrated approach is more efficient than separate curve estimation and the two-step approach. The integrated approach also provides more interpretable principle component functions in an application of estimating weekly wind power curves of a wind turbine.

Keywords: B-splines, functional data analysis, monotone smoothing, penalization, relative curvature function, spline smoothing
2010 MSC: 62G05, 62H25

1. Introduction

In this paper we consider the problem of estimating a collection of monoton curves. Estimation of a group of curves has been studies in the functional data analysis literature but existing methods are not directly applicable to estimating curves with shape constraints. We have the following three considerations in our situation. First, the underlying curves to be estimated are smooth and strictly-monotone. The monotonicity is often assumed on smooth curves such as cumulative distribution functions, survival functions, growth curves and so on. Second, a collection of curves instead of a single curve are of interest. The hope is that estimating these curves together can borrow strength across curves and do better than estimating each curve separately. Lastly, curves are observed with noises on an irregular and sparse grid. Assuming complete observation of curves on an equally-spaced grid is too restrictive in reality.

Our study is motivated from an application in industrial engineering where estimation of the *power curve* of a wind turbine is needed. The power curve explains the functional relationship between wind power output and wind speed input (Ackermann and Söder, 2005). It is especially useful for forecasting power production from a wind turbine (Ding, 2019). As illustrated in Figure 1, a power curve is theoretically smooth and monotonically increasing since a wind turbine produces higher power as wind speed increases. However, because of measurement errors or other environmental factors that possibly affect the power production, the data are noisy version of the smooth power curves. Moreover, because wind blows disorderly, the observed values of the input variable, wind speed, are irregularly spaced, and each power curve may have a different observed range. Figure 2 shows two examples of observed wind power curve based on weekly observations.

[*] Corresponding author. Address: Department of Statistics, 447 Blocker, 3143 TAMU, College Station, TX 77843-3143, USA. Phone: (979) 845-3141. Fax: (979) 845-3144.
Email address: lzhou@stat.tamu.edu. (Lan Zhou)

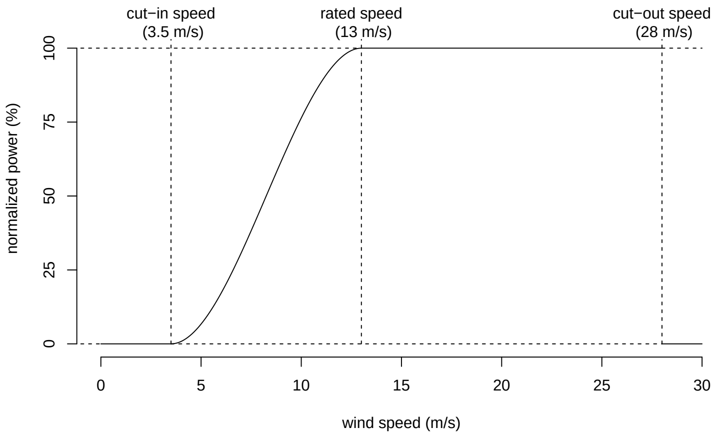


Figure 1: A nominal wind power curve. A turbine starts power production at the cut-in speed, reaches its full operation at the rated speed, and stops producing power at and beyond the cut-out speed. Power outputs are normalized by the rated power.

Estimation of single monotone curves has been studied extensively in the statistics literature. Existing spline-based approaches either use constrained coefficients estimation or use constrained optimization techniques; see Ramsay (1988), Kelly and Rice (1990), Zhang (2004), and Pya and Wood (2015). Through a reparametrization, Ramsay (1998) developed another spline-based approach that is constraint-free on spline coefficients and does not rely on constrained optimization. Local polynomial kernel methods for estimating single monotone curves include Hall and Huang (2001), Hall and Müller (2003) and Mammen and Yu (2007).

Ramsay (1998) proposed to estimate the so-called relative curvature curve instead of a monotone curve directly. His approach has an advantage of converting the problem of estimating a constrained function into that of estimating an unconstrained function. Nevertheless, that approach is designed to estimate a single monotone curve while we aim to estimate a collection of monotone curves sharing similar shapes. In particular, we would like to borrow strength across different curves during estimation. For example, when a curve is only partially-observed as in Figure 2(b), details of the curve when reaching the rated power output at high wind speed would be hardly recognized if the curve is estimated alone.

We thus aim at the joint estimation of a collection of monotone curves, rather than the one-by-one estimation. To that end, we make use of the concept of functional principal component analysis (fPCA), which is broadly used to represent multiple curves by a few key functions: a mean function and several leading principal component functions. The individual characteristics of each curve can also be preserved through principal component scores. By doing so, an incompletely observed curve such as Figure 2(b) can borrow the information from entire data; therefore, its estimated curve would have a reasonable shape as other curves.

The traditional approach of fPCA is defined similarly as the classical multivariate principal component analysis (PCA) but merely a summation changes into an integration (Ramsay, 2006). See also Rao (1958), Besse and Ramsay (1986), Castro et al. (1986) and Jones and Rice (1992). However, this traditional approach is limited to the case that all curves are completely observed at an equally-spaced grid. Even though one can project the data on a common grid and then apply the traditional approach, but it is not the best way to utilize the data. To overcome the drawbacks of traditional fPCA, James et al. (2000), Rice and Wu (2001), Zhou et al. (2008) and Guo et al. (2015) developed spline-based approaches for sparsely and irregularly sampled curves. On the other hand, Besse et al. (1997), Staniswalis and Lee (1998) and Yao et al. (2005) proposed kernel-based approaches for functional data modeling on an irregular grid. In this paper, we adopt a spline-based approach as aforementioned and particularly the idea of the reduced-rank model that James et al. (2000) suggested. While most fPCA techniques are developed for general curves without any shape constraints, we in this paper propose a fPCA model for monotone-constrained curves that has not been

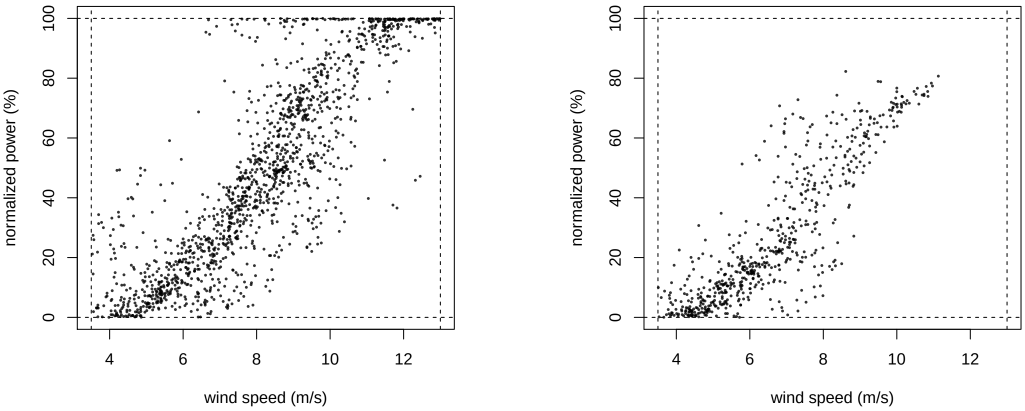


Figure 2: Weekly power curves from the cut-in speed to the rated speed (the strictly monotonically increasing range).

developed.

We study two approaches for the joint estimation of monotone curves via fPCA; we call them a *two-step* and an *integrated* approach, respectively. Both target the same structure in a functional model framework, however, the procedures for estimating unknown functions in the proposed model differ. The two-step approach simply performs existing two methods in a row; it first estimates a relative curvature for each monotone curve as suggested by Ramsay (1998), and then, the classical fPCA is applied to the estimated relative curvatures. On the other hand, the integrated approach is indeed a primary method we want to recommend, which estimates all of the spline coefficients in the proposed model simultaneously. Unlike the two-step approach, the integrated approach provides one unified algorithm.

The remainder of the paper is structured as follows. In Section 2, we review Ramsay (1998) to describe how to estimate a single monotone curve as well as its relative curvature and introduce the necessary background. In Section 3, we propose a fPCA model for a collection of monotone curves and develop two approaches to estimate model parameters. Section 4 presents a simulation study to compare the performances of several approaches: the Ramsay's approach, the two-step approach and the integrated approach. All approaches are applied to estimate power curves for a wind turbine in Section 5. The R code for producing the numerical results in the paper are available at `https://github.com/syeeun/jointmono`.

2. Estimation of a Monotone Curve

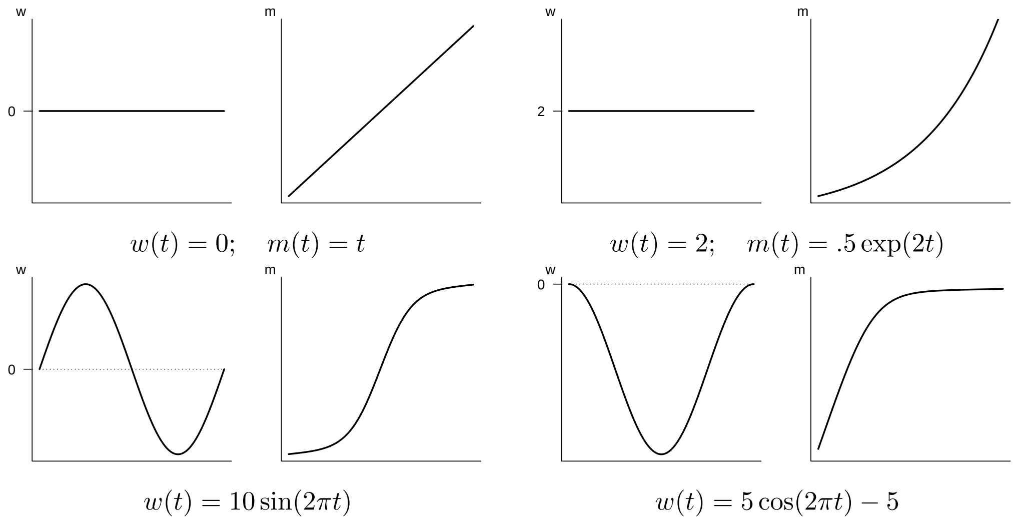
Consider the problem of estimating a function which belongs to the *class of monotone curves* $\mathcal{M}$ that consists of functions m that satisfy the following conditions,

1. $\log Dm$ is differentiable;
2. $D \log Dm = D^2m/Dm$ is Lebesgue square integrable,

where D^r refers to a differential operator of order r. These conditions ensure that m is strictly monotonically increasing ($-m$ is strictly monotonically decreasing) and its first derivative is smooth and finite almost everywhere. Ramsay (1998) showed that the functions in this class can be represented by a simple linear differential equation as

$$m = \beta_0 + \beta_1 D^{-1} \exp D^{-1} w, \tag{1}$$

where β_0 and β_1 are arbitrary constants and w is a Lebesgue square integrable function such that $w = D^2m/Dm$. The function w can be interpreted as the *relative curvature* of m, i.e., the size of the curvature



$w(t) = 0; \quad m(t) = t$ $\qquad$ $w(t) = 2; \quad m(t) = .5\exp(2t)$

$w(t) = 10\sin(2\pi t)$ $\qquad$ $w(t) = 5\cos(2\pi t) - 5$

Figure 3: Examples of relative curvatures w and monotone curves m. See how monotone curves look like according to their corresponding relative curvatures.

D^2m relative to the slope Dm. The equation (1) can also be written explicitly using integrals as

$$m(t) = \beta_0 + \beta_1 \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} w(u)\,\mathrm{d}u\,\mathrm{d}s. \tag{2}$$

See Ramsay (1998) and Ramsay (2006) for details about this monotone function representation.

We would like to point out that the relative curvature w in (1) and (2) indicates the particular shape of monotone curve m. Figure 3 illustrates four examples of monotone curves and their relative curvatures. In the case of constant relative curvatures as in (a) and (b), their explicit forms of m are available; zero $w(t) = 0$ leads a straight line $m(t) = t$, while non-zero constant $w(t) = c$ corresponds to an exponentially increasing curve $m(t) = 1/c\ \exp(ct)$. More complicated shapes of monotone curves can also be represented by a certain form of relative curvature curves as shown in (c) and (d); there are no explicit forms of m, though. In addition, when w is continuous, inflection points, at where a curve changes from concave to convex or vice versa, can be found by solving $w(t) = 0$. We set $\beta_0 = 0$ and $\beta_1 = 1$ in the figure to describe curves in a clear way.

Now we turn to the estimation problem. For monotone curve $m \in \mathcal{M}$, suppose its noisy observations satisfy

$$y(t_j) = m(t_j) + \epsilon_j, \qquad j = 1, \ldots, n, \tag{3}$$

where ϵ_j is a zero-mean random noise, and n is the total number of observations. Denote the vector of observations by $\boldsymbol{Y} = (y(t_1), \ldots, y(t_n))$. According to Ramsay (1998), an estimate of m is obtained by maximizing the penalized least squares criterion

$$F_\lambda(\boldsymbol{Y}|\beta_0, \beta_1, w) = n^{-1}\sum_{j=1}^{n}\{y(t_j) - m(t_j)\}^2 + \lambda \int_{\tau_0}^{\tau_1} w^2(t)\,\mathrm{d}t, \tag{4}$$

where λ is a smoothing parameter, and τ_0 and τ_1 are the lower and upper limit of t. Here, the roughness penalty is applied to the relative curvature function $w = m''/m'$, not to the second derivative m''. This ensures the smoothness of the relative curvature function w as well as the smoothness of the unknown monotone curve m. See Section 3 of Ramsay (1998) for more discussions.

Instead of estimating m directly, we estimate the relative curvature function w through basis expansion. Specifically, w is represented as a member of a q-dimensional space of spline functions, such that $w(t) =$

$\boldsymbol{b}(t)^T\boldsymbol{\theta}$, where $\boldsymbol{b}(t) = (b_1(t), \dots, b_q(t))^T$ is a vector of spline basis functions and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_q)^T$ is a vector of spline coefficients. In general, the basis expansion is only an approximation. However, when w is a smooth function, the approximation is good if a sufficiently large q is used (De Boor, 2001).

Following the spline basis expansion of w, the monotone curve m is represented as

$$m(t) = \beta_0 + \beta_1 \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \boldsymbol{b}(u)^T\boldsymbol{\theta} \, \mathrm{d}u \, \mathrm{d}s, \tag{5}$$

and hence, the problem of estimating the monotone function m becomes the problem of estimating parameters β_0, β_1 and $\boldsymbol{\theta}$. The fitting criterion (4) can be written in a vector-matrix form as

$$F_\lambda(\boldsymbol{Y}|\beta_0, \beta_1, \boldsymbol{\theta}) = n^{-1} \sum_{j=1}^{n} \left\{ y(t_j) - \beta_0 - \beta_1 \int_{\tau_0}^{t_j} \exp \int_{\tau_0}^{s} \boldsymbol{b}(u)^T\boldsymbol{\theta} \, \mathrm{d}u \, \mathrm{d}s \right\}^2 + \lambda \, \boldsymbol{\theta}^T \boldsymbol{\Omega} \boldsymbol{\theta}, \tag{6}$$

where $\boldsymbol{\Omega} = \int_{\tau_0}^{\tau_1} \boldsymbol{b}(t)\boldsymbol{b}(t)^T \, \mathrm{d}t$. To minimize the criterion (6), an iterative Fisher scoring procedure is performed for the basis coefficients $\boldsymbol{\theta}$, and β_0 and β_1 are updated by ordinary linear regression at each iteration. The smoothing parameter λ may be chosen by cross-validation techniques. For more details about this fitting algorithm, see section 3.1 of Ramsay (1998). With the estimated parameters $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\boldsymbol{\theta}}$, the fitted monotone curve is

$$\hat{m}(t) = \hat{\beta}_0 + \hat{\beta}_1 \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \hat{w}(u) \, \mathrm{d}u \, \mathrm{d}s,$$

where $\hat{w}(t) = \boldsymbol{b}(t)^T\hat{\boldsymbol{\theta}}$.

3. Joint Estimation of Monotone Curves

Our study primarily aims at estimating a collection of monotone curves, rather than a single monotone curve. Section 3.1 proposes a functional model that can represent a collection of monotone curves. Two approaches for estimating the proposed model are presented in Section 3.2 and Section 3.3.

3.1. Modelling a collection of monotone curves

Consider a collection of M functions in the class of monotone curves, $\{m_i \in \mathcal{M} \mid i = 1, \dots, M\}$. Suppose observations of a function m_i have the same structure as (3), while each m_i is observed at a possibly different set of t in a common fixed interval $[\tau_0, \tau_1]$. Precisely, an observation of m_i at t_{ij} is given by

$$y_i(t_{ij}) = m_i(t_{ij}) + \epsilon_{ij}, \qquad j = 1, \dots, n_i, \tag{7}$$

where n_i denotes the number of observations for the i-th curve. Following (refeqn:ramsay2), each m_i is expressed via a relative curvature function w_i such that

$$m_i(t) = \beta_{0i} + \beta_{1i} \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} w_i(u) \, \mathrm{d}u \, \mathrm{d}s, \tag{8}$$

where τ_0 is a common lower limit of integration. In practice, it is natural to choose τ_0 and τ_1 respectively as the minimum and maximum of the observation points t_{ij}'s from all M functions.

3.2. Two-step approach

The two-step approach literally estimates model parameters in the following two steps.

First, we fit a single monotone curve model (5) for each curve by applying the approach described in Section 2. Then, we obtain a collection of individually-fitted monotone curves denoted as

$$\tilde{m}_i(t) = \hat{\beta}_{0i} + \hat{\beta}_{1i} \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \tilde{w}_i(u) \, \mathrm{d}u \, \mathrm{d}s,$$

where $\tilde{w}_i(t) = \boldsymbol{b}(t)^T\tilde{\boldsymbol{\theta}}_i$ with the estimated parameters $\hat{\beta}_{0i}$ and $\hat{\beta}_{1i}$, and the estimated basis coefficients $\tilde{\boldsymbol{\theta}}_i$ for $i \in \{1, \ldots, M\}$.

Second, we discretize each of the fitted relative curvature functions $\{\tilde{w}_i \mid i = 1, \ldots, M\}$ on a sufficiently dense grids $(\nu_1, \ldots, \nu_G)$ in the range (τ_0, τ_1); that is $\tau_0 \leq \nu_1 < \ldots < \nu_G \leq \tau_1$ for a moderate size G and $\nu_{j+1} - \nu_j = \nu_j - \nu_{j-1}$ for any $j \in \{2, \ldots, G-1\}$. Then, we treat the *augmented* data, denoted by $\{\tilde{w}_i(\nu_j) \mid j = 1, \ldots, G; i = 1, \ldots, M\}$, as if they were equidistantly and completely observed data. We finally fit the augmented data with the *reduced rank model* proposed in James et al. (2000).

To be specific about the reduced rank model, we assume that for a fixed i, $\{\tilde{w}_i(\nu_j) \mid j = 1, \ldots, G\}$ are observed trajectories of a smooth function w_i, that is

$$\tilde{w}_i(\nu_j) = w_i(\nu_j) + \zeta_{ij}, \tag{9}$$

where $\zeta_{ij} \sim \mathcal{N}(0, \xi^2)$, and w_i is represented by a mean function $\mu(t)$ plus a linear combination of a common set of functions $\boldsymbol{f}(t) = \{f_1(t), \ldots, f_K(t)\}^T$, with its own set of coefficients $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iK})^T$;

$$w_i(t) = \mu(t) + \boldsymbol{f}(t)^T\boldsymbol{\alpha}_i. \tag{10}$$

For identifiability of the representation (10), the orthonormality of $\boldsymbol{f}(t)$ is required such that

$$\int \boldsymbol{f}(t)\,\boldsymbol{f}(t)^T\,\mathrm{d}t = \boldsymbol{I}_K, \tag{11}$$

where $\boldsymbol{I}_K$ is a $K \times K$ identity matrix (i.e. $\int f_k f_l = 0$ and $\int f_k^2 = 1$ for all $k \neq l \in \{1, \ldots, K\}$). It is also assumed that the coefficients are drawn from a multivariate normal distribution that has a mean zero and a diagonal covariance matrix with decreasing diagonal components; $\boldsymbol{\alpha}_i \sim (0, \Sigma)$. We call $\boldsymbol{f}(t)$ the *principal component functions* and $\boldsymbol{\alpha}_i$ the *principal component scores.* A large enough K ensures the needed flexibility in representing unknown relative curvature functions.

However, in our setting the principal component functions are not pre-specified and need to be determined by the data. To this end, we suppose that these principal component functions fall in a low-dimensional subspace of a function space spanned by a set of B-spline functions, $\boldsymbol{b}(t) = \{b_1(t), \ldots, b_q(t)\}^T$ with $q \gg K$, such that

$$\mu(t) = \boldsymbol{b}(t)^T\boldsymbol{\theta}_\mu; \quad \boldsymbol{f}(t) = \boldsymbol{b}(t)^T\boldsymbol{\Theta}_f, \tag{12}$$

where $\boldsymbol{\theta}_\mu$ is a $q \times 1$ vector and $\boldsymbol{\Theta}_f = (\boldsymbol{\theta}_{f_1}, \ldots, \boldsymbol{\theta}_{f_K})^T$ is a $q \times K$ matrix of basis coefficients. A large enough q ensures the needed flexibility in representing the unknown functions. Furthermore, we restrict that the basis functions are linearly independent and standardized, $\int \boldsymbol{b}(t)\boldsymbol{b}(t)^T\,\mathrm{d}t = \boldsymbol{I}_q$, and the coefficient matrix to be orthonormal, $\boldsymbol{\Theta}_f^T\boldsymbol{\Theta}_f = \boldsymbol{I}_K$. Such restrictions guarantee the orthonormality of the principal component functions in (11) as

$$\int \boldsymbol{f}(t)\boldsymbol{f}(t)^T\mathrm{d}t = \boldsymbol{\Theta}_f^T \int \boldsymbol{b}(t)\boldsymbol{b}(t)^T\mathrm{d}t\ \boldsymbol{\Theta}_f = \boldsymbol{\Theta}_f^T\boldsymbol{\Theta}_f = \boldsymbol{I}_K.$$

In the simulation and data application studies of later sections, the B-splines are used and the creation of orthonormal B-splines follows the procedure in Appendix A.

Denote $\tilde{\boldsymbol{w}}_i = \{\tilde{w}_i(\nu_1), \ldots, \tilde{w}_i(\nu_G)\}^T$ and $\boldsymbol{\zeta}_i = \{\zeta_i(\nu_1), \ldots, \zeta_i(\nu_G)\}^T$, then the model for the augmented data (9) gets the vector-matrix form as

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{b}^T\boldsymbol{\theta}_\mu + \boldsymbol{b}^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i + \boldsymbol{\zeta}_i, \tag{13}$$

where $\boldsymbol{b} = \{\boldsymbol{b}(\nu_1), \ldots, \boldsymbol{b}(\nu_G)\}$ is a $q \times G$ matrix of basis function values evaluated at the fine grids $(\nu_1, \ldots, \nu_G)$. To estimate the unknown parameters in the model (13), EM algorithm is used since $\boldsymbol{\alpha}$ is unobservable and hence treated as a missing variable (Dempster et al., 1977).

Finally, plugging in the estimated coefficients $\hat{\boldsymbol{\theta}}_\mu$ and $\hat{\boldsymbol{\Theta}}_f$, we get $\hat{\mu}(t) = \boldsymbol{b}(t)^T\hat{\boldsymbol{\theta}}_\mu$ and $\hat{\boldsymbol{f}}(t) = \boldsymbol{b}(t)^T\hat{\boldsymbol{\Theta}}_f$. Compute the conditional expectation $\hat{\boldsymbol{\alpha}}_i = \mathrm{E}(\boldsymbol{\alpha}_i|\tilde{\boldsymbol{w}}_i)$ for $i \in \{1, \ldots, M\}$. Then, the fitted monotone function via the two-step approach for each $i \in \{1, \ldots, M\}$ is

$$\hat{m}_i(t) = \hat{\beta}_{0i} + \hat{\beta}_{1i}\int_{\tau_0}^{t} \exp\int_{\tau_0}^{s} \left\{\hat{\mu}(u) + \hat{\boldsymbol{f}}(u)^T\hat{\boldsymbol{\alpha}}_i\right\}\mathrm{d}u\,\mathrm{d}s,$$

where the parameters $\hat{\beta}_{0i}$ and $\hat{\beta}_{1i}$ are estimated at the first step.

3.3. Integrated approach

The integrated approach is inspired by the two-step approach, however, it does not fit the augmented data but fit the observed data directly. In other words, it unifies the two steps of Section 3.2 so that all the parameters are estimated together, not step-by-step.

The integrated approach is based on the model specified by (7), (8), (10), and (12), while the principal component scores are treated as fixed effects that satisfy $(1/M)\sum_{i=1}^{M}\boldsymbol{\alpha}_i = \mathbf{0}$. Combining (8) and (10), our model for the ith monotone curve is

$$m_i(t) = \beta_{0i} + \beta_{1i}\int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \{\mu(u) + \boldsymbol{f}(u)^T\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s, \tag{14}$$

for $i = 1, \ldots, M$. With the basis expansions of μ and $\boldsymbol{f}$ given in (12), we represent (14) as

$$m_i(t) = \beta_{0i} + \beta_{1i}\int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \{\boldsymbol{b}(u)^T\boldsymbol{\theta}_\mu + \boldsymbol{b}(u)^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s, \tag{15}$$

for $i = 1, \ldots, M$. For simplicity of notations, we define

$$h_i(t) = h(t;\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i) = \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \{\boldsymbol{b}(u)^T\boldsymbol{\theta}_\mu + \boldsymbol{b}(u)^T\boldsymbol{\Theta}_f\boldsymbol{\alpha}_i\}\,\mathrm{d}u\,\mathrm{d}s,$$

and denote the observed data as $\boldsymbol{Y}_i = \{y_i(t_1), \ldots, y_i(t_{n_i})\}^T$. Accordingly, (7) and (15) can be represented in the vector-matrix form

$$\boldsymbol{Y}_i = \beta_{0i}\mathbf{1}_{n_i} + \beta_{1i}\boldsymbol{H}_i(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i) + \boldsymbol{\epsilon}_i, \tag{16}$$

where $\mathbf{1}_{n_i}$ is an $n_i \times 1$ vector of ones, $\boldsymbol{H}_i(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i) = \{h_i(t_{i1}), \ldots, h_i(t_{in_i})\}^T$, $\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i$ are unknown parameters, and $\boldsymbol{\epsilon}_i = \{\epsilon_{i1}, \ldots, \epsilon_{in_i}\}^T$.

Inspired by (4), we estimate the unknown parameters by minimizing the following penalized scaled sum of squared residuals

$$F_{\lambda_\mu}(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\sigma^2}\sum_{i=1}^{M} ||\boldsymbol{Y}_i - \beta_{0i}\mathbf{1}_{n_i} - \beta_{1i}\boldsymbol{H}_i(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i)||^2 + \lambda_\mu \int \mu^2(t)\,\mathrm{d}t, \tag{17}$$

where $\boldsymbol{\beta} = \{(\beta_{01}, \ldots, \beta_{0M})^T, (\beta_{11}, \ldots, \beta_{1M})^T\}$ is a set of intercepts and slopes, $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M\}^T$ is an $M \times K$ matrix of principal component scores, and λ_μ is a penalty parameters. Note that the penalty term $\int \mu^2(t)\,\mathrm{d}t$ extends the penalty $\int w^2(t)\,\mathrm{d}t$ in the penalized least squares criterion (4) for single curve estimation (Ramsay, 1998). Since the relative curvature function $w_i(t) = m_i''(t)/m_i'(t)$ measures the smoothness of the monotone curve m_i, $\mu(t) = (1/M)\sum_{i=1}^{M} w_i(t)$ can be interpreted as a measure of average smoothness. Moreover, similar to $\int w^2(t)\,\mathrm{d}t$, the penalty $\int \mu^2(t)\,\mathrm{d}t$ has the effect of keeping fitted curves away from the boundary condition $m_i'(t) = 0$ for all i. If $m_i'(t) = 0$ for some i, then $\mu(t) = (1/M)\sum_{i=1}^{M} m_i''(t)/m_i'(t) = \infty$.

The penalty term in (17) also helps ensure computational stability of the Fisher scoring algorithm (Appendix B). Using the orthonormality of the basis functions $\boldsymbol{b}(t)$, we obtain

$$\int \mu(t)^2\,\mathrm{d}t = \boldsymbol{\theta}_\mu^T \int \boldsymbol{b}(t)\boldsymbol{b}(t)^T\,\mathrm{d}t\ \boldsymbol{\theta}_\mu = \boldsymbol{\theta}_\mu^T\boldsymbol{\theta}_\mu. \tag{18}$$

Without this penalty term, the cross-product matrix of gradient vectors used in the algorithm for updating $\boldsymbol{\theta}_\mu$ is numerically close to being singular, causing instability of the algorithm (see Appendix C), while the penalty effectively adds $\lambda_\mu\boldsymbol{I}$ to this cross-product matrix and solves the problem. The singularity issue also occurs when updating $\boldsymbol{\Theta}_f$ in the Fisher scoring algorithm, and a natural solution is to add a penalty $\sum_{k=1}^{K}\boldsymbol{\theta}_{f_k}^T\boldsymbol{\theta}_{f_k}$ to (17). Therefore, the final minimizing objective function for estimating the parameters is

$$F_{\lambda_\mu,\lambda_f}(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{\sigma^2}\sum_{i=1}^{M} \|\boldsymbol{Y}_i - \beta_{0i}\mathbf{1}_{n_i} - \beta_{1i}\boldsymbol{H}_i(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i)\|^2 + \lambda_\mu\boldsymbol{\theta}_\mu^T\boldsymbol{\theta}_\mu + \lambda_f\sum_{k=1}^{K}\boldsymbol{\theta}_{f_k}^T\boldsymbol{\theta}_{f_k}. \tag{19}$$

The orthonormality constraint on $\boldsymbol{\Theta}_f$, $\boldsymbol{\Theta}_f^T\boldsymbol{\Theta}_f = \boldsymbol{I}_K$, implies that $\sum_{k=1}^K \boldsymbol{\theta}_{f_k}^T\boldsymbol{\theta}_{f_k} = K$. Thus, the second penalty term in (19) does not depend on unknown parameters to be estimated, so it does not have the regularization effect as the first penalty term. It cannot be dropped from (19) because $\sum_{k=1}^K \boldsymbol{\theta}_{f_k}^T\boldsymbol{\theta}_{f_k} = K$ may not be satisfied in some steps of the Fisher scoring algorithm. The inclusion of the second penalty term is mainly for numerical stability of the computational algorithm.

Details of the iterative Fisher scoring algorithm to minimize (19) is given in Appendix 5. See the next subsection for details about the choice of penalty parameters.

Denote the estimated parameter obtained by minimizing (19) as $\hat{\boldsymbol{\theta}}_\mu$, $\hat{\boldsymbol{\Theta}}_f$, and $\hat{\boldsymbol{\alpha}}_i, \hat{\beta}_{0i}, \hat{\beta}_{1i}$ for $i \in \{1, \ldots, M\}$. For each $i \in \{1, \ldots, M\}$, the fitted monotone function via the integrated approach is given by

$$\hat{m}_i(t) = \hat{\beta}_{0i} + \hat{\beta}_{1i}\int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \left\{\hat{\mu}(u) + \hat{\boldsymbol{f}}^T(u)\hat{\boldsymbol{\alpha}}_i\right\} \mathrm{d}u\, \mathrm{d}s,$$

where $\hat{\mu}(t) = \boldsymbol{b}(t)^T\hat{\boldsymbol{\theta}}_\mu$ and $\hat{\boldsymbol{f}}(t) = \boldsymbol{b}(t)^T\hat{\boldsymbol{\Theta}}_f$.

3.4. Model selection

Specification of B-splines

The number of knots and the positions of the knots are not crucial in many applications as long as sufficiently many knots are placed densely, since the roughness penalty helps regularize the estimation and prevent overfitting; see also Eilers and Marx (1996). A moderate number of equidistant knots over the data range, typically 10-20 knots, is often sufficient.

Choice of penalty parameters, λ_μ and λ_f

For each fixed value of K, the 5-fold (within function) cross-validation is used to choose the two penalty parameters. Observations from each curve are randomly divided into 5 groups of equal size. Each group is set aside once as a validation set while other 4 groups are used as a training set to fit the model. The fitted model is then applied to the validation set to compute the cross-validation sum of squared errors. The 5 such sums of squared errors are then summed up to obtain the overall 5-fold cross-validated sum of squared errors.

One can use a commonly used search algorithm such as the Nelder-Mead simplex method (Nelder and Mead, 1965) to minimize the 5-fold cross-validated sum of squared errors. For all examples of simulation and application in the following sections, we employed a straightforward grid-search on a 12×12 grid in log-scale for each penalty parameter. We found that our grid search performed competitively to the Nelder-Mead, which usually required more function evaluations than the grid search. Moreover, the grid search method does not need to specify initial values as the Nelder-Mead does, whose performance varies with the initial values.

We found that the cross-validation criterion is not sensitive to the choice of λ_f. This is not surprising, since as we discussed following (19), the corresponding penalty term is mainly used for numerical stability of the algorithm and does not introduce regularization on function estimation.

Choice of the number of principal components, $\hat{K}$

For a fixed K, let $\hat{\lambda}_\mu(K), \hat{\lambda}_f(K)$ denote the penalty parameters chosen by the 5-fold cross-validation and let $\mathrm{CV}_K(\hat{\lambda}_\mu(K), \hat{\lambda}_f(K))$ denote the corresponding cross-validation (CV) sum of squared errors. We choose K by minimizing the CV sum of squared errors, i.e.,

$$\hat{K} = \mathrm{argmin}_K \mathrm{CV}_K(\hat{\lambda}_\mu(K), \hat{\lambda}_f(K)).$$

The performance of CV in selecting the significant K is illustrated in Section 4 using a simulation study.

Alternatively, K can be chosen by dropping the components whose scores have a relatively small variance compared with the variance of the preceding component; this idea is similar to Cattell's scree test (Cattell, 1966). Specifically, one can fit the model with a sufficient number of principal component functions, and plot the variances of principal component scores in a decreasing order. From this plot, K can be chosen where the variance curve makes an elbow toward less steep decline. We later describe how to perform this procedure for a practical example in Section 5.

4. Simulation Study

We compared our proposed integrated approach for joint estimation of monotone functions with the two-step approach and Ramsay's single curve method in a simulation study. We implemented Ramsay's method according to the algorithm given in Ramsay (1998). When applying the integrated approach and Ramsay's method, we used quadratic splines with 7 equally-spaced interior knots on $[0, 1]$; this corresponds to $q = 10$. When applying the two-step approach, the number of grid points was set to be $G = 1000$.

4.1. Simulation setup

Without loss of generality, we assume that all curves have zero intercepts and equal slopes (i.e. $\beta_0 = 0$ and $\beta_1 = 1$). We generate $M = 50$ monotone curve trajectories from the functional model

$$y_i(t_{ij}) = \int_0^{t_j} \exp \int_0^s w_i(u)\,\mathrm{d}u\,\mathrm{d}s + \epsilon_{ij},$$

for $i \in \{1, \ldots, 50\}$ and $j \in \{1, \ldots, n_i\}$, where n_i is an integer uniformly sampled between 50 and 100 and t_{ij}'s are uniformly distributed in $[0, 1]$. The mean function is $\mu(t) = 5 - 10t$, and two principal component functions (i.e. $K = 2$) are $f_1(t) = \sqrt{2}\sin(2\pi t)$ and $f_2(t) = \sqrt{2}\cos(2\pi t)$. Therefore, the relative curvature functions are $w_i(t) = \mu(t) + f_1(t)\alpha_{1i} + f_2(t)\alpha_{2i}$. The principal component scores are generated according to

$$\begin{pmatrix}\alpha_{1i}\\ \alpha_{2i}\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.5^2 & 0\\ 0 & 0.1^2\end{pmatrix}\right).$$

We simulate the errors ϵ_{ij} from two types of distributions, $\sigma \times \mathcal{N}(0, 1)$ or $\sigma \times t(4)$, with three noise levels, $\sigma \in \{0.01, 0.05, 0.1\}$. Here, $t(4)$ denotes the t distribution with 4 degrees of freedom.

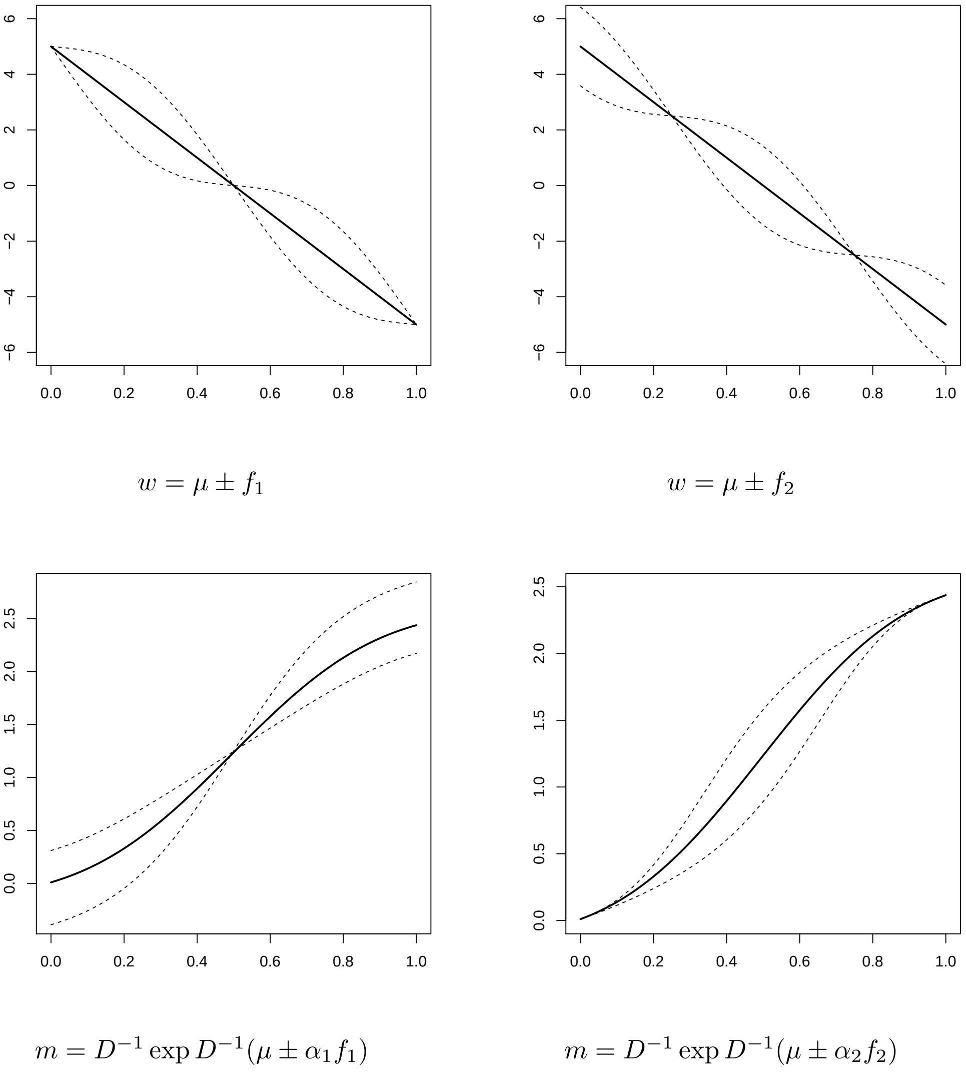
Figure 4 shows the mean function μ and how the principal components functions affect the relative curvature functions as well as the response curves in our simulation. It can be seen that f_1 represents the variation at the boundary, and also provides how much the curvature changes before and after an inflection point for each curve. On the other hand, f_2 represents the variation at the center of the range, so any response curves with relatively large variation in the middle might have large values of scores α_{2i}.

4.2. Assessment criteria

We assess the estimators of w and m using the mean integrated squared error (MISE), integrated squared bias (I-sq-bias), and integrated variance (I-var). For the ith curve, let $\hat{w}_i^l(t)$ denote the estimate of $w_i(t)$ at the lth simulation run, $1 \le l \le L$. These assessment criteria for w_i are defined as

$$\begin{aligned}
\mathrm{MISE}(w_i) &= \frac{1}{L}\sum_{l=1}^{L}\int_{\tau_0}^{\tau_1}\{\hat{w}_i^l(t) - w_i(t)\}^2\,\mathrm{d}t,\\
\text{I-sq-bias}(w_i) &= \int_{\tau_0}^{\tau_1}\{\bar{\hat{w}}_i(t) - w_i(t)\}^2\,\mathrm{d}t,\\
\text{I-var}(w_i) &= \frac{1}{L}\sum_{l=1}^{L}\int_{\tau_0}^{\tau_1}\{\hat{w}_i^l(t) - \bar{\hat{w}}_i(t)\}^2\,\mathrm{d}t,
\end{aligned}$$

where $\bar{\hat{w}}_i(t) = (1/L)\sum_{l=1}^{L}\hat{w}_i^l(t)$ is the average over L simulation runs, where integrations can be evaluated as a Riemann sum. The criteria used for m_i are defined similarly. For the integrated approach, since the $\hat{w}_i$ has a basis expansion using an orthonormal basis, the evaluation of the integrals for assessing estimation quality of w_i can be simplified using the equality $\int\{\boldsymbol{b}(t)^T\boldsymbol{\theta}\}^2\,\mathrm{d}t = \boldsymbol{\theta}_i^T\boldsymbol{\theta}_i$ where $\boldsymbol{b}(t)$ is an orthnomal basis.



$w = \mu \pm f_1$

$w = \mu \pm f_2$

$m = D^{-1} \exp D^{-1}(\mu \pm \alpha_1 f_1)$

$m = D^{-1} \exp D^{-1}(\mu \pm \alpha_2 f_2)$

Figure 4: Illustrations of the relative curvature function w (top) and the corresponding monotone response curve m (bottom) in the simulation study. The mean and principal component functions are $\mu(t) = 5 - 10\,t$, $f_1(t) = \sqrt{2}\sin(2\pi t)$, and $f_2(t) = \sqrt{2}\cos(2\pi t)$. In (c) and (d), $\alpha_1 = 1$ and $\alpha_2 = 2$ are chosen to clearly schematize the curves.

Table 1: Summary of a simulation study for the choice of the number of principal component functions K where the data were simulated with $K = 2$.

	$\sigma = 0.01$			$\sigma = 0.05$			$\sigma = 0.1$		
optimal K	1	2	3	1	2	3	1	2	3
occurrence	0%	100%	0%	0%	99%	1%	0%	94%	6%

To assess the overall quality for estimating M curves, we take average to arrive at the following metrics

$$\begin{aligned} \text{MISE} &= \frac{1}{M}\sum_{i=1}^{M} \text{MISE}(w_i), \\ \text{Bias}^2 &= \frac{1}{M}\sum_{i=1}^{M} \text{I-sq-bias}(w_i), \\ \text{SD}^2 &= \frac{1}{M}\sum_{i=1}^{M} \text{I-var}(w_i), \end{aligned}$$

which are used to compare different methods in our simulation study. Note that the summation of Bias^2 and SD^2 equals MISE. Evaluation criteria for m_i's can be defined similarly.

4.3. Result summary

For the integrated approach, we estimated parameters for each simulation run with $K = 1, 2$, and 3. The cross-validation (CV) errors are then compared to choose the optimal K on each simulation. The results based on $L = 500$ simulation runs are summarized in Table 1. The data-generating number of principal functions, $K = 2$, was selected 100% times when $\sigma = 0.01$, 99% when $\sigma = 0.05$ and 94% when $\sigma = 0.1$. As the noise level increases, the chance that $K = 3$ is selected increases, because it gets harder for the method to distinguish noise and signals.

Table 2 summarizes the bias, standard deviation (SD; square root of integrated variance), and MISE of $\hat{m}$ and $\hat{w}$ estimated from the three approaches over $L = 500$ runs for two types of error distributions ($\sigma \times \mathcal{N}(0, 1)$ and $\sigma \times t(4)$) and three noise levels ($\sigma = 0.01, 0.05$, and 0.10). We have the following observations.

- For estimating the monotone function m, our integrated method clearly outperforms the other two methods in terms of absolute bias, SD, and MISE, at all noise levels. For both Ramsay's method and our integrated method, the SD dominates the absolute bias, while for the two-step method, SD and the absolute bias have similar magnitude. This indicates that the two-step method actually introduces more bias in estimating m.

- For estimating the relative curvature function w, our integrated method is also a clear winner in terms of absolute bias, SD, and MISE. Its improvement over the other two methods is substantial. For all three methods, the SD dominates the absolute bias in magnitude and it contributes as the major part of the MISE. The two-step method has slightly higher bias but lower SD than Ramsay's method; this is expected, because the two-step method applies a principal components reduction to the $\hat{w}$'s obtained by the Ramsay's method. The principal components reduction naturally reduces variance with the cost of introducing bias.

- For each type of error distributions, as the noise level increases, the SDs and MISEs increase for all methods, while there is no clear pattern for the bias.

- The comparison results when the error distribution is a scale multiple of t distribution are similar to those when the error distribution is a normal distribution. The main difference between the case of t distributed errors and the corresponding case of normally distributed errors is that the MISE

Table 2: Summary of simulation results for the estimates of monotone curves ($\hat{m}$) and relative curvatures ($\hat{w}$) for $\sigma = 0.01, 0.05$, and 0.10 using the Ramsay, the two-step, and the integrated approaches. Error distributions are a scale multiple of either $\mathcal{N}(0,1)$, the standard normal distribution, or $t(4)$, the t distribution with 4 degrees of freedom. |Bias| is the absolute value of the bias, SD is the standard deviation, and MISE is the mean integrated squared error; the summary statistics for $\hat{m}$ were multiplied by 100 for making clear the difference in numbers.

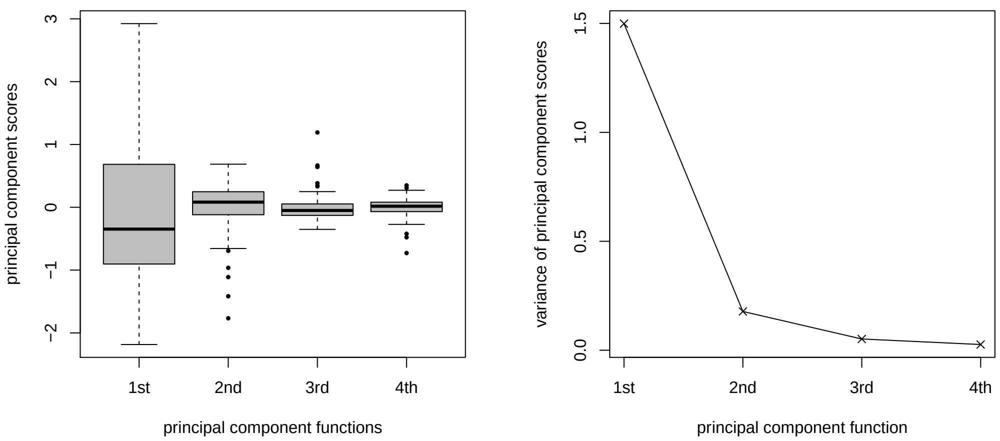
Error distribution			Ramsay		Two-step		Integrated	
	σ		$\hat{m}$	$\hat{w}$	$\hat{m}$	$\hat{w}$	$\hat{m}$	$\hat{w}$
$\sigma \times \mathcal{N}(0,1)$	0.01	|Bias|	0.10	2.01	2.58	2.03	0.03	0.50
		SD	0.36	4.51	1.54	4.36	0.16	0.63
		$\sqrt{\text{MISE}}$	0.37	4.94	3.01	4.81	0.16	0.81
	0.05	|Bias|	0.28	1.67	3.17	1.85	0.06	0.34
		SD	1.31	11.99	3.20	10.46	0.81	2.15
		$\sqrt{\text{MISE}}$	1.34	12.11	4.51	10.62	0.81	2.18
	0.1	|Bias|	0.66	1.92	6.78	2.13	0.08	0.38
		SD	2.46	17.02	10.73	14.06	1.65	5.19
		$\sqrt{\text{MISE}}$	2.54	17.13	10.75	14.22	1.65	5.21
$\sigma \times t(4)$	0.01	|Bias|	0.09	1.85	2.64	1.88	0.03	0.41
		SD	0.45	5.40	1.51	5.24	0.23	0.83
		$\sqrt{\text{MISE}}$	0.46	5.71	3.04	5.57	0.23	0.92
	0.05	|Bias|	0.40	1.59	3.28	1.85	0.07	0.32
		SD	1.78	13.79	4.53	11.73	1.19	3.09
		$\sqrt{\text{MISE}}$	1.82	13.89	5.60	11.88	1.19	3.11
	0.1	|Bias|	0.94	2.24	6.98	2.35	0.15	0.80
		SD	3.27	19.44	77.79	16.68	2.43	8.08
		$\sqrt{\text{MISE}}$	3.40	19.57	78.10	16.75	2.43	8.12

has increased for all methods. This is expected since a t-distribution has heavier tails than a normal distribution with the same scale parameter.

5. Application: Wind Power Curve Data

We fitted the power curves that originally motivated our study as introduced in Section 1, using the Ramsay's, the two-step, and the integrated approaches. In the dataset, wind power productions and wind speeds were recorded every 10-minute for about one and a half years. We assumed that one week records create one curve; hence, the total number of curves is $M = 74$. We only considered the range of wind speeds, from 4 to 12 m/s, in which most wind curves are strictly increasing as shown in Figure 1. We used $G = 100$ grid points for the two-step approach, and used quadratic splines with 7 equally-spaced interior knots for Ramsay's approach and the integrated approach, corresponding to $q = 10$ basis functions. Ramsay's approach was used in the first step of the two-step approach.

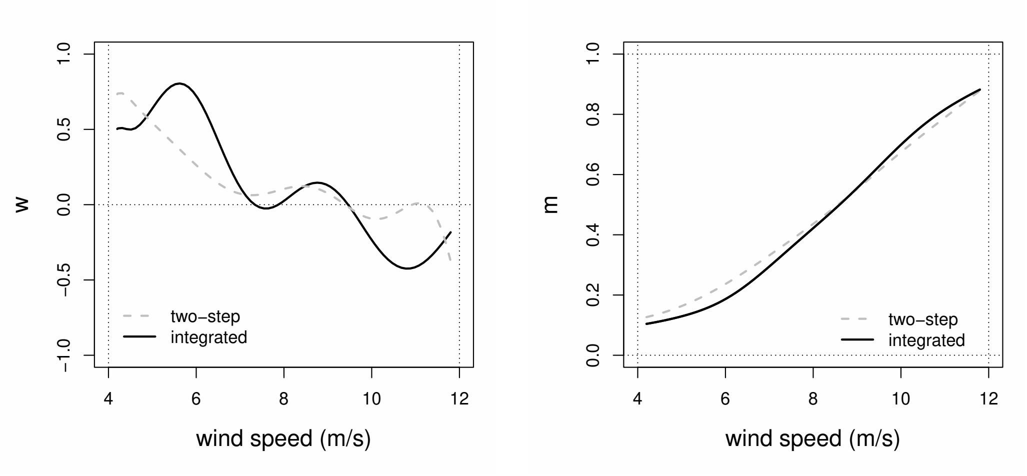
For the integrated approach, the tuning parameters λ_μ and λ_f were determined using the 5-fold cross validation as described in Section 3.4, and chosen in the grid ranges of $\log_{10}(\lambda_\mu) = -8, \ldots, -1$ and $\log_{10}(\lambda_f) = 1, \ldots, 6$. The number of significant principal component functions was selected as $\hat{K} = 2$ using the 5-fold cross-validation. The scree plot shown in Figure 5 also suggests the choice of $\hat{K} = 2$. The minimum 5-fold CV sum of squared errors for the integrated approach were 3107.13, 3057.36, 3136.85, and 3073.42 for $K = 1, 2, 3, 4$, respectively. These can be compared with the 5-fold CV sum of squared errors of 3817.44 of the Ramsay's approach. Note that the same partition of the data were used when calculating the 5-fold CV errors for all methods.



Distribution of PC scores

Variance of PC scores

Figure 5: Distribution and variance of principal component scores. Two principal component functions ($K = 2$) are sufficient to explain the overall variation of data.



$\mu(t)$

$m(t) = \beta_0 + \beta_1 \int \exp \int \mu(t)\,\mathrm{d}t\,\mathrm{d}t$

Figure 6: (a) The fitted mean relative curvature curve $\mu(t)$ and (b) the corresponding monotone curve when β_0 and β_1 are fixed at the average of all individual curves, using the two-step approach (dashed lines) and the integrated approach (solid lines).

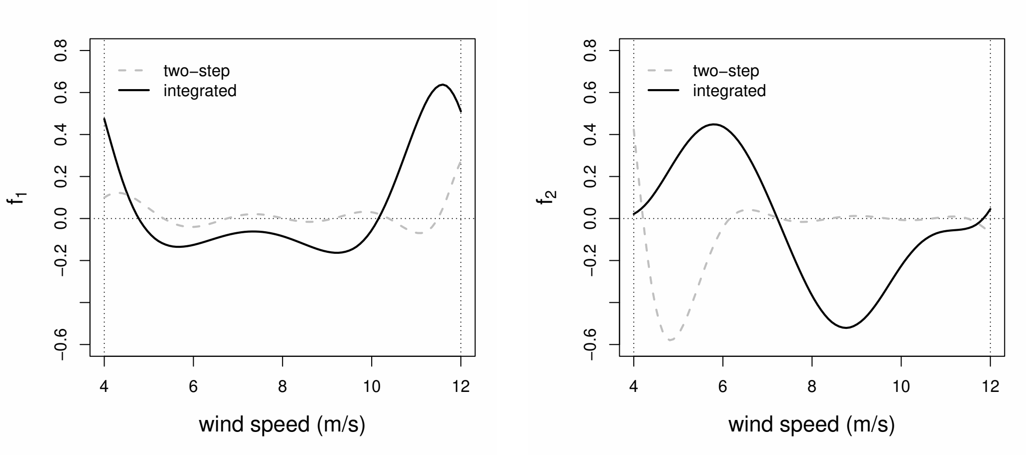


Figure 7: The estimated first two principal component functions using the two-step approach (dashed lines) and the integrated approach (solid lines).

Figure 6 illustrates the fitted mean function of relative curvatures, $\mu(t)$, and its corresponding monotone curve, $m(t)$, in (a) and (b), respectively, using the two-step and integrated approaches. From $\mu(t)$ estimated by the integrated approach, we observe a change of curvature at around 7 m/s of wind speed. The change of sign of $m''(t)$ from positive to negative indicates that the power curve changes from a convex increasing function to a concave increasing function. The estimated curves by the two-step approach show similar patterns, but the curvature is close to zero after 7 m/s of wind speed.

Figure 7 shows the estimated first two principal component functions by the integrated approach and the two-step approach. The first principal component function by the integrated approach explains the contrast of curvature of the power curve between the boundaries and the middle of the wind speed range. The magnitude of scores on this component describes how much the power curve accelerates at the low or high ends of the range. The second principal component function obtained by the integrated approach explains the contrast of curvature before and after the inflection point at around 7 m/s of wind speed. As comparison, the principal component functions obtained by the two-step approach are much harder to interpret. The first principle component function is close to zero in the middle of the wind speed range, while the second principle component function is close to zero in the range of 6–12 m/s of wind speed. In fact the principle component functions obtained by the two-step approach have very large values around the boundaries (not shown in the figure), which suggests that these principle component functions may have captured the uninteresting boundary effects of the first step nonparametric curve fitting, instead of our interest in variation of power curves.

Acknowledgement

Part of Yei Eun Shin's work was done while she was a doctoral student at Department of Statistics, Texas A&M University.

References

Ackermann, T., Söder, L., 2005. Wind power in power systems: an introduction, in: Wind Power in Power Systems. Wiley Online Library, pp. 25–51.

Besse, P., Ramsay, J.O., 1986. Principal components analysis of sampled functions. Psychometrika 51, 285–311.

Besse, P.C., Cardot, H., Ferraty, F., 1997. Simultaneous non-parametric regressions of unbalanced longitudinal data. Computational Statistics & Data Analysis 24, 255–270.

Castro, P.E., Lawton, W.H., Sylvestre, E.A., 1986. Principal modes of variation for processes with continuous sample curves. Technometrics 28, 329–337.

Cattell, R.B., 1966. The scree test for the number of factors. Multivariate behavioral research 1, 245–276.

De Boor, C., 2001. Calculation of the smoothing spline with weighted roughness measure. Mathematical Models and Methods in Applied Sciences 11, 33–41.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) 39, 1–38.

Ding, Y., 2019. Data Science for Wind Energy. Chapman and Hall/CRC.

Eilers, P.H.C., Marx, B.D., 1996. Flexible smoothing with b-splines and penalties. Statistical Science 11, 89–102.

Guo, M., Zhou, L., Huang, J.Z., Härdle, W.K., 2015. Functional data analysis of generalized regression quantiles. Statistics and Computing 25, 189–202.

Hall, P., Huang, L.S., 2001. Nonparametric kernel regression subject to monotonicity constraints. Annals of Statistics 29, 624–647.

Hall, P., Müller, H.G., 2003. Order-preserving nonparametric regression, with applications to conditional distribution and quantile function estimation. Journal of the American Statistical Association 98, 598–608.

James, G.M., Hastie, T.J., Sugar, C.A., 2000. Principal component models for sparse functional data. Biometrika 87, 587–602.

Jones, M.C., Rice, J.A., 1992. Displaying the important features of large collections of similar curves. The American Statistician 46, 140–145.

Kelly, C., Rice, J., 1990. Monotone smoothing with application to dose-response curves and the assessment of synergism. Biometrics 46, 1071–1085.

Longford, N., 1987. A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. Biometrika 1987, 817–827.

Mammen, E., Yu, K., 2007. Additive isotone regression. IMS Lecture Notes - Monograph Series 55, 179–195.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. The Computer Journal 7, 308–313.

Pya, N., Wood, S.N., 2015. Shape constrained additive models. Statistics and Computing 25, 543–559.

Ramsay, J.O., 1988. Monotone regression splines in action. Statistical science 3, 425–441.

Ramsay, J.O., 1998. Estimating smooth monotone functions. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60, 365–375.

Ramsay, J.O., 2006. Functional data analysis. Wiley Online Library.

Rao, C.R., 1958. Some statistical methods for comparison of growth curves. Biometrics 14, 1–17.

Rice, J.A., Wu, C.O., 2001. Nonparametric mixed effects models for unequally sampled noisy curves. Biometrics 57, 253–259.

Staniswalis, J.G., Lee, J.J., 1998. Nonparametric regression analysis of longitudinal data. Journal of the American Statistical Association 93, 1403–1418.

Yao, F., Müller, H.G., Wang, J.L., 2005. Functional data analysis for sparse longitudinal data. Journal of the American Statistical Association 100, 577–590.

Zhang, J.T., 2004. A simple and efficient monotone smoother using smoothing splines. Journal of Nonparametric Statistics 16, 779–796.

Zhou, L., Huang, J.Z., Carroll, R.J., 2008. Joint modelling of paired sparse functional data using principal components. Biometrika 95, 601–619.

Appendix A. Creation of Orthonormal Basis Functions

We follow the computation of creating orthonormal basis function from Zhou et al. (2008). They provided details about the transformation from arbitrary basis functions to orthonormal. Here we explain briefly their techniques.

Let $\tilde{\boldsymbol{b}}(t) = \{\tilde{b}_1(t), \dots, \tilde{b}_q(t)\}^T$ be an initially chosen general B-splines; this is not necessarily orthonormal. A transformation matrix $\boldsymbol{T}$ such that $\boldsymbol{b}(t) = \boldsymbol{T}\tilde{\boldsymbol{b}}(t)$ can be constructed as follows. Write $\tilde{\boldsymbol{b}} = \{\tilde{\boldsymbol{b}}(t_1), \dots, \tilde{\boldsymbol{b}}(t_g)\}^T$ for the equally-spaced and sufficiently dense grid, $(t_1, \dots, t_g)$. Apply the QR decomposition to $\tilde{\boldsymbol{b}} = \boldsymbol{QR}$, where $\boldsymbol{Q}$ has orthonormal columns and $\boldsymbol{R}$ is an upper triangular matrix. Then, $\boldsymbol{T} = (g/L)^{1/2}\boldsymbol{R}^{-T}$, where $L = t_g - t_1$, will be a desirable transformation matrix since

$$\frac{L}{g}\boldsymbol{b}^T\boldsymbol{b} = \frac{L}{g}\boldsymbol{T}\tilde{\boldsymbol{b}}^T\tilde{\boldsymbol{b}}\boldsymbol{T}^T = \frac{L}{g}\boldsymbol{T}\boldsymbol{R}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{R}\boldsymbol{T}^T = \boldsymbol{I}.$$

See Figure 8 that illustrates an example of the transformation.

Appendix B. Fisher Scoring Algorithm for the Integrated Approach

This section specifies the algorithm of minimizing the penalized likelihood (19) for the estimation of unknown parameters, described in the model (15).

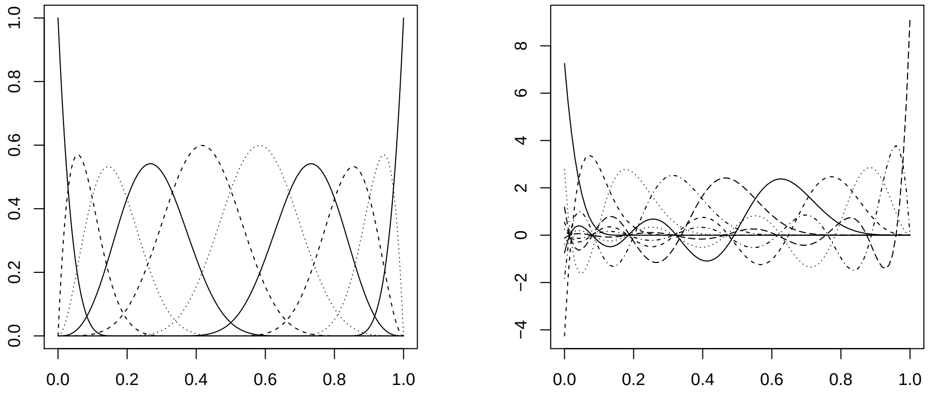


Figure 8: Illustration of B-spline functions where $q = 10$ with an order 5; (Left) B-splines; (Right) orthonormalized B-splines

We want to point out that the integrated approach treats principal component scores as fixed effects, which satisfy the following conditions

$$\sum_{i=1}^{M} \alpha_{ik} = 0, \quad \forall k, \qquad \sum_{i=1}^{M} \alpha_{i1}^2 > \ldots > \sum_{i=1}^{M} \alpha_{iK}^2,$$

for identifiable individual-level characteristics among the scores. By doing so, we can avoid highly complicated computations caused by two integrals and an exponential between those integrals. It is impossible to derive a closed form of conditional distributions of $\boldsymbol{\alpha}$ unlike the two-step approach, which has a linear form as in (13). Guo et al. (2015) also carried out the parameter estimation by considering random effects as fixed effects.

Denote $\boldsymbol{H}_i = \boldsymbol{H}_i(\boldsymbol{\theta}_\mu, \boldsymbol{\Theta}_f, \boldsymbol{\alpha}_i)$. The non-linear maximum likelihood equations for $\boldsymbol{\theta}_\mu, \boldsymbol{\theta}_{f_1}, \ldots, \boldsymbol{\theta}_{f_K}$ and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M$ are obtained by taking a partial differentiation to the criterion (19) for each parameter

$$\begin{aligned}
0 = \frac{\partial F}{\partial \boldsymbol{\theta}_\mu} &= -2 \sum_{i=1}^{M} \beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu} \boldsymbol{r}_i / \sigma^2 + 2\lambda_\mu \boldsymbol{\theta}_\mu, \\
0 = \frac{\partial F}{\partial \boldsymbol{\theta}_{f_k}} &= -2 \sum_{i=1}^{M} \beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \boldsymbol{r}_i / \sigma^2 + 2\lambda_f \boldsymbol{\theta}_f, \quad \forall k \in \{1, \ldots, K\} \\
0 = \frac{\partial F}{\partial \boldsymbol{\alpha}_i} &= -2\beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i} \boldsymbol{r}_i / \sigma^2, \quad \forall i \in \{1, \ldots, M\}
\end{aligned}$$

where $\boldsymbol{r}_i := \boldsymbol{Y}_i - \beta_{0i} \mathbf{1}_{n_i} - \beta_{1i} \boldsymbol{H}_i$ is an $n_i \times 1$ vector of residuals. The columns of the partial derivative matrices $\partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_\mu (q \times n_i), \partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_{f_k} (q \times n_i)$ and $\partial \boldsymbol{H}_i / \partial \boldsymbol{\alpha}_i (K \times n_i)$ are respectively the following vectors evaluated at $t_1, \ldots, t_{n_i}$,

$$\begin{aligned}
\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu} &= \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \boldsymbol{b}(u) \{ \boldsymbol{b}(u)^T \boldsymbol{\theta}_\mu + \boldsymbol{b}(u)^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i \} \, \mathrm{d}u \, \mathrm{d}s, \\
\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_{f_k}} &= \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \alpha_{ik} \boldsymbol{b}(u) \{ \boldsymbol{b}(u)^T \boldsymbol{\theta}_\mu + \boldsymbol{b}(u)^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i \} \, \mathrm{d}u \, \mathrm{d}s = \alpha_{ik} \frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu}, \\
\frac{\partial h_i(t)}{\partial \boldsymbol{\alpha}_i} &= \int_{\tau_0}^{t} \exp \int_{\tau_0}^{s} \boldsymbol{\Theta}_f^T \boldsymbol{b}(u) \{ \boldsymbol{b}(u)^T \boldsymbol{\theta}_\mu + \boldsymbol{b}(u)^T \boldsymbol{\Theta}_f \boldsymbol{\alpha}_i \} \, \mathrm{d}u \, \mathrm{d}s = \boldsymbol{\Theta}_f^T \frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu}.
\end{aligned}$$

One only needs to calculate the integrals once because the latter two terms $\partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_{f_k}$ and $\partial \boldsymbol{H}_i / \partial \boldsymbol{\alpha}_i$ are expressed as a product of certain coefficients and $\partial \boldsymbol{H}_i / \partial \boldsymbol{\theta}_\mu$.

The iterative procedure of Fisher scoring algorithm (Longford, 1987) below can be used to find the solution of the above equations:

1. Initialize $\beta_{0i}^0, \beta_{1i}^0, \boldsymbol{\theta}_\mu^0, \boldsymbol{\Theta}_f^0, \boldsymbol{\alpha}_i^0$.
2. Update $\boldsymbol{\theta}_\mu^l$ as
$$\boldsymbol{\theta}_\mu^l \leftarrow \boldsymbol{\theta}_\mu^{l-1} + \Big[\sum_{i=1}^M (\beta_{1i}^l)^2 \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu}^T + \lambda_\mu \boldsymbol{I}_q \Big]^{-1} \Big[\sum_{i=1}^M \beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_\mu} \boldsymbol{r}_i - \lambda_\mu \boldsymbol{\theta}_\mu \Big].$$
3. Update $\boldsymbol{\theta}_{f_k}^l$ for $\forall k \in \{1, \ldots, K\}$ as
$$\boldsymbol{\theta}_{f_k}^l \leftarrow \boldsymbol{\theta}_{f_k}^{l-1} + \Big[\sum_{i=1}^M (\beta_{1i}^l)^2 \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}}^T + \lambda_f \boldsymbol{I}_q \Big]^{-1} \Big[\sum_{i=1}^M \beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\theta}_{f_k}} \boldsymbol{r}_i - \lambda_f \boldsymbol{\theta}_f \Big],$$
and re-update by orthonormalized ones through QR decomposing $\boldsymbol{\Theta}_f^l = \{\boldsymbol{\theta}_{f_1}^l, \ldots, \boldsymbol{\theta}_{f_k}^l\}^T$.
4. Update $\boldsymbol{\alpha}_i^l$ for $\forall i \in \{1, \ldots, M\}$ as
$$\boldsymbol{\alpha}_i^l \leftarrow \boldsymbol{\alpha}_i^{l-1} + \Big[(\beta_{1i}^l)^2 \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i}^T \Big]^{-1} \Big[\beta_{1i} \frac{\partial \boldsymbol{H}_i}{\partial \boldsymbol{\alpha}_i} \boldsymbol{r}_i \Big],$$
and rearrange such that $\sum_{i=1}^M \alpha_{ik} = 0$ for $\forall k$ and $\sum \alpha_{i1}^2 > \ldots > \sum \alpha_{iK}^2$.
5. Update β_{0i}^l and β_{1i}^l by estimates of a linear regression as
$$\boldsymbol{Y}_i \sim \beta_{0i} + \beta_{1i} \boldsymbol{H}_i(\boldsymbol{\theta}_\mu^l, \boldsymbol{\Theta}_f^l, \boldsymbol{\alpha}_i^l),$$
for $\forall i \in \{1, \ldots, M\}$.
6. Iterate step 2 to 5 until all converge.

Once the algorithm converges, an estimate of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^M ||\boldsymbol{Y}_i - \hat{\beta}_{0i} \mathbf{1}_{n_i} - \hat{\beta}_{1i} \boldsymbol{H}_i(\hat{\boldsymbol{\theta}}_\mu, \hat{\boldsymbol{\Theta}}_f, \hat{\boldsymbol{\alpha}}_i)||^2,$$

where $\hat{\beta}_{0i}, \hat{\beta}_{1i}, \hat{\boldsymbol{\theta}}_\mu, \hat{\boldsymbol{\Theta}}_f, \hat{\boldsymbol{\alpha}}_i$ denote the converged estimates of the above iterative algorithm.

Appendix C. Computational Singularity of the Cross-product Matrix in the Fisher Scoring Algorithm

The partial differentials in the equations form the majority part of the cross-product matrix of gradient vectors used in the Fisher scoring algorithm. We here address the importance of penalization not just for smoothing but also for computational stability.

Consider the partial differential of $h_i(t)$ with respect to $\boldsymbol{\theta}_\mu$, that is

$$\frac{\partial h_i(t)}{\partial \boldsymbol{\theta}_\mu} = \int \boldsymbol{B}(t) \exp\{\boldsymbol{b}(t)^T \boldsymbol{\theta}_\mu + \boldsymbol{b}(t)^T \boldsymbol{\theta}_f \boldsymbol{\alpha}_i\} \, \mathrm{d}t = \int \boldsymbol{B}(t) \exp\{W(t)\} \, \mathrm{d}t,$$

where $\boldsymbol{B}(t) = \int \boldsymbol{b}(t) \, \mathrm{d}t$ is a q-vector of integrated basis functions, and $W(t) = \int w(t) \, \mathrm{d}t = \boldsymbol{B}(t)^T \boldsymbol{\theta}_\mu + \boldsymbol{B}(t)^T \boldsymbol{\theta}_f \boldsymbol{\alpha}_i$ is an integrated function of relative curvature w. Then, $\partial h_i(t) / \partial \boldsymbol{\theta}_\mu$ is a vector of functions coming from integral of exponential (> 0), multiplied by integrated basis function, $\boldsymbol{B}(t)$. Since the $\boldsymbol{B}(t)$ looks like the left panel of Figure 9, the function elements have therefore different shapes to each other; some increase fast

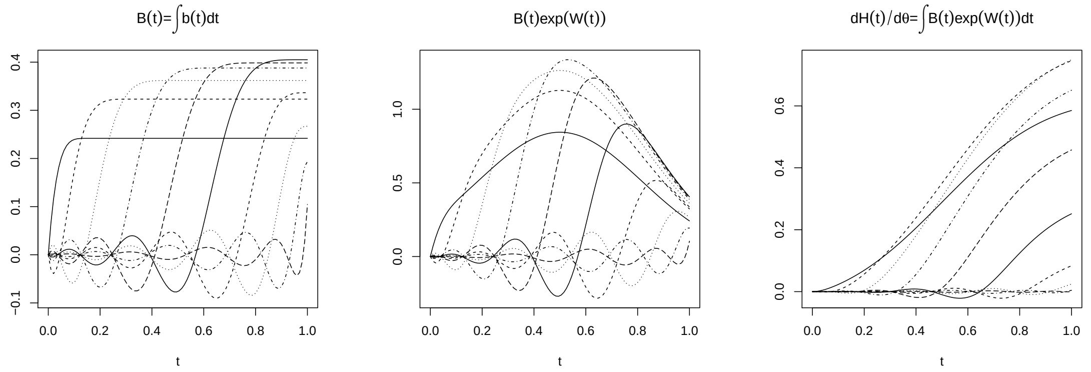


Figure 9: (left) integrated basis; (center) intermediate function in calculation; (right) partial differential of $H(t)$

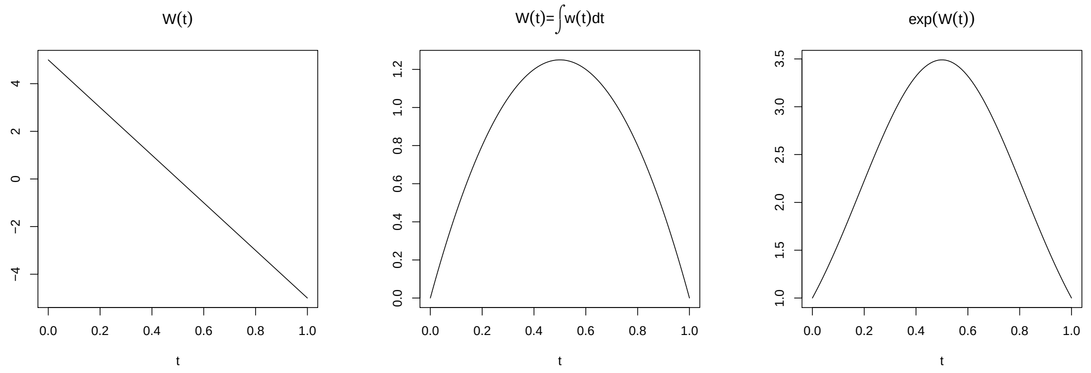


Figure 10: An example of $w(t)$, its integrated form, and the exponential of the integrated.

and some are always near zero. For this reason, when these functions are evaluated at observed points, the cross-product matrix of gradient vectors, $(\partial\boldsymbol{H}_i/\partial\boldsymbol{\theta}_\mu)(\partial\boldsymbol{H}_i/\partial\boldsymbol{\theta}_\mu)^T$, will be computationally singular because of relatively near zero values at its diagonal.

To help understanding, we illustrate the form of functions derived at each computation step. Suppose there is a curve of $w(t)$, which is a straight line, as illustrated in Figure 10. Accordingly, the form of partial differential of $h(t)$ can be drawn by multiplying $\boldsymbol{B}(t)$ (the left panel of Figure 9) and $W(t)$ (the center panel of Figure 10); see the center and right panel of Figure 9. As aforementioned, functions of $\partial h_i(t)/\partial\boldsymbol{\theta}_\mu$ have different forms, therefore the evaluated cross product matrix has values, for this example, as shown in Table 3.

To summarize, the penalty parameters λ_μ and λ_f play a role for computational stability for the algorithm. If there is an issue with non-existence of inverse matrix due to computationally singularity, setting λ's at suitable values will be a key technique to make the algorithm converge.

Table 3: An example of a cross product matrix corresponding to $\boldsymbol{b}$ in Figure 8 and w in Figure 10 to illustrate the necessity of ridge regularization. Relatively too small values at the lower right corner of the diagonal causes numerical singularity.

	bspl5.1	bspl5.2	bspl5.3	bspl5.4	bspl5.5	bspl5.6	bspl5.7	bspl5.8	bspl5.9	bspl5.10
bspl5.1	123813.68	154995.62	148755.14	117527.56	68236.01	26109.53	3875.89	522.57	-54.09	36.31
bspl5.2	154995.62	194311.94	187005.93	148283.47	86438.78	33183.68	4943.97	663.89	-67.48	45.55
bspl5.3	148755.14	187005.93	181096.78	144939.39	85369.10	33047.57	4969.21	660.28	-63.89	43.82
bspl5.4	117527.56	148283.47	144939.39	118140.88	71347.62	28169.83	4324.97	561.59	-48.24	34.47
bspl5.5	68236.01	86438.78	85369.10	71347.62	45340.78	18968.78	3025.46	397.59	-35.91	25.25
bspl5.6	26109.53	33183.68	33047.57	28169.83	18968.78	9039.44	1698.93	187.24	-2.23	5.50
bspl5.7	3875.89	4943.97	4969.21	4324.97	3025.46	1698.93	558.72	47.87	7.38	-2.06
bspl5.8	522.57	663.89	660.28	561.59	397.59	187.24	47.87	39.03	-9.00	4.30
bspl5.9	-54.09	-67.48	-63.89	-48.24	-35.91	-2.23	7.38	-9.00	6.41	-2.85
bspl5.10	36.31	45.55	43.82	34.47	25.25	5.50	-2.06	4.30	-2.85	1.37