Computational Statistics and Data Analysis 146 (2020) 106925

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Design optimal sampling plans for functional regression models

Hyungmin Rha [a], Ming-Hung Kao [a], Rong Pan [b,*]

[a] Department of Statistics, Arizona State University, Tempe, AZ, 85281, USA
[b] Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ, 85281, USA

ARTICLE INFO

Article history:
Received 11 June 2019
Received in revised form 16 January 2020
Accepted 17 January 2020
Available online 5 February 2020

Keywords:
Functional data analysis
Functional linear model
Functional principal components
Longitudinal data

ABSTRACT

Functional regression models are widely considered in practice. To make a precise statistical inference, a good sampling schedule for collecting informative functional data is needed. However, there has not been much research on the optimal sampling schedule design for functional regression model so far. To address this design issue, an efficient computational approach is proposed for generating the best sampling plan in the function-on-function linear regression setting. The obtained sampling plan allows a precise estimation of the predictor function and a precise prediction of the response function. The proposed approach can also be applied to identify the optimal sampling plan for the problem with scalar-on-function linear regression model. Through case studies, this approach is demonstrated to outperform the methods proposed in the previous studies.

1. Introduction

Functional data analysis has wide applications such as in brain imaging (Jiang et al., 2009; Saleh et al., 2017; Zhu et al., 2018), pharmacokinetics (Lai et al., 2006), and marketing (Sood et al., 2009). Observations that are obtained repeatedly over time in functional data analysis can either be on a regular dense time grid (Ferraty, 2014) or on a sparse irregular time grid (Peng and Paul, 2009). It is known that measuring functional data on a dense, regular grid by, e.g., a high frequency recording machine, has several advantages for data analytics (Rice, 2004). However, collecting such a data set can be expensive and may not always be possible in practice due to limited resource and/or practical constraints. In such a situation, a judicious selection of sampling time points is important for yielding high statistical efficiency (Pan et al., 2019). Several previous studies have discussed optimal sampling times when functional data are measured at sparse and irregularly spaced time points. For example, Wu et al. (2018) discussed the optimal sampling schedules to capture between-subject variability. Ji and Müller (2017) attempted to find the optimal sampling designs for recovering the unknown trajectory of the underlying predictor function and for predicting a scalar response through a functional linear model. Park et al. (2018) worked in the same direction but aimed for the design that maximizes the sum of two optimality criteria, one for recovering predictor function and the other for predicting a scalar response.

Following these works, we are concerned with the optimal sampling schedules for recovering the trajectory of a predictor function and a response. But here, we allow the response to either be either a scalar or a function. We also consider the flexible weighted sum criterion for finding the compound optimal design that strikes the balance between the statistical efficiencies in recovering the functional predictor and predicting the response based on user-specified weights.

[*] Corresponding author.
E-mail address: rong.pan@asu.edu (R. Pan).

https://doi.org/10.1016/j.csda.2020.106925
0167-9473/© 2020 Elsevier B.V. All rights reserved.

To find such a design, we first extend the results of Ji and Müller (2017), and Park et al. (2018) to derive our optimality criteria for a function-on-function regression model. This extension is also presented as an open research problem in the unpublished Ph.D. thesis of Li (2017). Here, we rigorously build our results on the theories developed by Yao et al. (2005b), and in a sense, our derived optimality criteria can be viewed as a pragmatic approximation to that of Li (2017). Our criteria can be used directly in a search algorithm for obtaining optimal designs. We then propose a probabilistic subset search (PSS) algorithm to efficiently find single- or multi-objective optimal designs for both function-on-function and scalar-on-function regression models. In simulation studies, we show that our proposed algorithm outperforms some existing approaches.

This paper is organized as follows. In Section 2, we introduce the basic notations and our derived optimality criteria for recovering the predictor function and predicting the response function. In Section 3, we discuss the implementations of some computational methods and introduce a new algorithm. Simulation studies are presented in Section 4. We then apply the proposed approach to some real applications in Section 5. A discussion can be found in Section 6.

2. Statistical framework

Let $(X_i(\cdot), Y_i(\cdot))$ be a pair of smooth but unobservable random trajectories for the ith subject. Throughout the paper, $X_i(\cdot)$ denotes the predictor function and $Y_i(\cdot)$ represents the response function. They are defined over continuous compact domains, $\mathcal{S}$ and $\mathcal{T}$, respectively, and have unknown smooth mean functions, $E(X(s)) = \mu_X(s)$ and $E(Y(t)) = \mu_Y(t)$, continuous covariance surfaces, $\text{Cov}(X(s_1), X(s_2)) = \Gamma_X(s_1, s_2)$ and $\text{Cov}(Y(t_1), Y(t_2)) = \Gamma_Y(t_1, t_2)$, and continuous cross-covariance surface, $\text{Cov}(X(s), Y(t)) = C_{XY}(s, t)$. We also assume that $X(s)$ and $Y(t)$ are square integrable random processes. Then, the Mercer's Theorem applies with $\Gamma_X(s_1, s_2) = \sum_{m=1}^{\infty} \rho_m \psi_m(s_1)\psi_m(s_2)$, and $\Gamma_Y(t_1, t_2) = \sum_{k=1}^{\infty} \lambda_k \phi_k(t_1)\phi_k(t_2)$, where ψ_m is the mth eigenfunction of the covariance operator of $X(s)$ with corresponding eigenvalue ρ_m, $\rho_1 \geq \rho_2 \geq \cdots \geq 0$, and ϕ_k is the kth eigenfunction for $Y(t)$ with eigenvalue λ_k, $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$. With mild conditions (Hsing et al., 2015), we can write

$$\begin{aligned} X_i(s) &= \mu_X(s) + \sum_{m=1}^{\infty} \zeta_{im}\psi_m(s), \quad \text{and} \\ Y_i(t) &= \mu_Y(t) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t), \end{aligned} \tag{1}$$

where for the ith subject, $\zeta_{im} = \int_{\mathcal{S}}(X_i(s) - \mu_X(s))\psi_m(s)ds$ and $\xi_{ik} = \int_{\mathcal{T}}(Y_i(t) - \mu_Y(t))\phi_k(t)dt$ are the functional principal component (FPC) scores for $X(s)$ and $Y(t)$, respectively. Note that the random variables, ζ_{im} (ξ_{ik}) have mean 0 and variance ρ_m (λ_k), and the FPC scores are uncorrelated. We define U_{il} as the (noisy) observation of $X_i(s_{il})$ at $s_{il} \in \mathcal{S}$, and V_{ij} as the observation for $Y_i(t_{ij})$ at $t_{ij} \in \mathcal{T}$; $l = 1, \ldots, L_i$, $j = 1, \ldots, J_i$ and $i = 1, \ldots, n$. In particular $U_{il} = X_i(s_{il}) + \varepsilon_{il}$ and $V_{ij} = Y_i(t_{ij}) + \epsilon_{ij}$, where ε_{il}'s and ϵ_{ij}'s are independent random noise with mean 0 and variances, σ_X^2 and σ_Y^2, respectively; see also Yao et al. (2005b). Note that for dimension reduction, the infinite sums in (1) are often approximated by respective finite sums using only the first few eigenfunctions and FPC scores.

To predict the unknown function $Y(\cdot)$ with a predictor function $X(\cdot)$, Yao et al. (2005b) considered the following function-on-function linear regression model in Ramsey and Silverman (2005),

$$E(Y(t)|X) = \mu_Y(t) + \int_{\mathcal{S}} \beta(s, t)X^c(s)ds = \mu_Y(t) + \sum_{k=1}^{\infty}\sum_{m=1}^{\infty} b_{km}\zeta_m\phi_k(t), \tag{2}$$

where $X^c(s) = X(s) - \mu_X(s)$ and $\beta(s, t)$ is the coefficient function that can be represented as $\beta(s, t) = \sum_{k=1}^{\infty}\sum_{m=1}^{\infty} b_{km}\psi_m(s)\phi_k(t)$ for $b_{km} = E(\zeta_m\xi_k)/E(\zeta_m^2)$; see He et al. (2000) and Yao et al. (2005b) for details. In practice, the unknown quantities in (1) and (2) need to be estimated. There is much work on estimations under the functional principal component analysis (FPCA) framework. Estimation methods for densely observed data were discussed in Castro et al. (1986) and Rice and Silverman (1991), and a shrinkage method for estimating the principal component scores was proposed in Yao et al. (2003). Estimation with sparse data was discussed by several authors using different approaches such as the mixed effects model (James et al., 2000), the local linear smoother (Yao et al., 2005b), the geometric approach (Peng and Paul, 2009), and the fast covariance estimation (Xiao et al., 2018). In this paper, we consider the FPCA function of 'fdapace' package (Dai et al., 2018) in R for estimation, although some other methods can also be considered. For a new subject with p observations $\boldsymbol{U} = (U(s_1), \ldots, U(s_p))^T$, we then have the following predictions of $X(s)$ and $Y(t)$, respectively (see also Yao et al. 2005a and Yao et al. 2005b):

$$\hat{X}^M(s) = \hat{\mu}_X(s) + \sum_{m=1}^{M} \hat{\zeta}_m\hat{\psi}_m(s); \quad \text{and} \quad \hat{Y}^{KM}(t) = \hat{\mu}_Y(t) + \sum_{k=1}^{K}\sum_{m=1}^{M} \hat{b}_{km}\hat{\zeta}_m\hat{\phi}_k(t),$$

where $\hat{\mu}_X(s)$ is the estimate of $\mu_X(s)$, $\hat{\mu}_Y(t)$ is an estimates of $\mu_Y(t)$, $\hat{\psi}_m(s)$ is the estimated mth eigenfunction for $X(s)$, $\hat{\phi}_k(t)$ is the estimated kth eigenfunction for $Y(t)$, $\hat{b}_{km}$ is the estimated coefficient of b_{km}, and $\hat{\zeta}_m = \hat{E}(\zeta_m|\boldsymbol{U})$ is the mth predicted FPC score; $m = 1, \ldots, M$ and $k = 1, \ldots, K$. The finite integers M for $X(s)$ and K for $Y(t)$ are selected so that the majority of variability of $X(s)$ and $Y(t)$ are captured; see Jiang et al. (2009). The asymptotic distributions of $\hat{X}^M(s)$ and $\hat{Y}^{KM}(t)$ are provided in Yao et al. (2005a,b), and

$$\begin{aligned} &\hat{X}^M(s) - X(s) \,\dot{\sim}\, N(0, \hat{\boldsymbol{\psi}}^M(s)^T(\hat{R} - \hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T)\hat{\boldsymbol{\psi}}^M(s)), \text{ and} \\ &\hat{Y}^{KM}(t) - E(Y(t)|X) \,\dot{\sim}\, N(0, \hat{\boldsymbol{\phi}}^K(t)^T\hat{B}_{KM}(\hat{R} - \hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T)\hat{B}_{KM}^T\hat{\boldsymbol{\phi}}^K(t)), \end{aligned} \tag{3}$$

where $\hat{R}$ is an $M \times M$ diagonal matrix whose diagonal elements are the estimated eigenvalues for $X(s)$, $\hat{\varGamma}_*$ is the estimated covariance matrix of $\boldsymbol{U}$, $\hat{\boldsymbol{\psi}}^M(s) = (\hat{\psi}_1(s), \ldots, \hat{\psi}_M(s))^T$, $\hat{\boldsymbol{\phi}}^K(t) = (\hat{\phi}_1(t), \ldots, \hat{\phi}_K(t))^T$, $\hat{B}_{KM}$ is a $K \times M$ matrix whose (k, m) element is $\hat{b}_{km}$ and $\hat{H}$ is the covariance matrix between the estimated FPC scores and the observations whose mth row is $\mathrm{Cov}(\hat{\zeta}_m, \boldsymbol{U})$. We aim at finding the best sampling schedule to sample the $X(s)$ for this new subject so that the recovery of the predictor function using $\hat{X}^M(s)$ and the prediction of the response function with $\hat{Y}^{KM}(t)$ are precise in some sense. To this end, we make use of (3) to derive the following optimality criteria for selecting a good sampling schedule $\mathbf{s} = (s_1, \ldots, s_p)^T$:

$$\hat{F}_X(\mathbf{s}) = \mathrm{tr}(\hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T), \text{ and} \tag{4}$$

$$\hat{F}_Y(\mathbf{s}) = \mathrm{tr}(\hat{B}_{KM}^T\hat{B}_{KM}\hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T). \tag{5}$$

We note that, by replacing $\hat{B}_{KM}$ with a $1 \times M$ vector whose mth element is $\hat{b}_m = \hat{E}(\zeta_m Y)/\hat{\rho}_m$, (5) is reduced to the criterion derived in Park et al. (2018) for a scalar response Y.

The derivations of (4) and (5) can be found in S.1 of the supplementary document. As mentioned there, $\mathrm{tr}(\hat{R})$, and $\mathrm{tr}(\hat{B}_{KM}^T\hat{B}_{KM}\hat{R})$ form upper bounds for $\hat{F}_X$ and $\hat{F}_Y$, respectively. We thus define the following relative efficiencies:

$$\begin{aligned} &\widehat{RE}_X(\mathbf{s}) = \frac{\mathrm{tr}(\hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T)}{\mathrm{tr}(\hat{R})}, \text{ and} \\ &\widehat{RE}_Y(\mathbf{s}) = \frac{\mathrm{tr}(\hat{B}_{KM}^T\hat{B}_{KM}\hat{H}\hat{\varGamma}_*^{-1}\hat{H}^T)}{\mathrm{tr}(\hat{B}_{KM}^T\hat{B}_{KM}\hat{R})}. \end{aligned} \tag{6}$$

As a direct consequence of the proof of Theorem 2 of Park et al. (2018), $\widehat{RE}_X(\mathbf{s})$ and $\widehat{RE}_Y(\mathbf{s})$ converge to 1 as $p \to \infty$.

One may also search for designs maximizing $\hat{F}_X$ and $\hat{F}_Y$, respectively, to provide approximations of sharper upper bounds to replace $\mathrm{tr}(\hat{R})$ and $\mathrm{tr}(\hat{B}_{KM}^T\hat{B}_{KM}\hat{R})$ in (6). Maximizing the sum of $\widehat{RE}_X$ and $\widehat{RE}_Y$ might strike a good trade-off between the two objectives. However, it might sometimes fail to meet the experimenter's needs when one of the objectives is more important than the other. To overcome this problem, we consider the weighted sum approach to allow a set of user-specified weights $W = \{w_1, \ldots, w_a\}$, where $0 \le w_i \le 1$, and w_i's are all distinct, and to find designs maximizing

$$w_i\widehat{RE}_X(\mathbf{s}) + (1 - w_i)\widehat{RE}_Y(\mathbf{s}), \quad i = 1, \ldots, a. \tag{7}$$

3. Search algorithms

We follow Ji and Müller (2017) and Park et al. (2018) to find the optimal design with p time points on a discretized domain that has N ($> p$) equally spaced points. We study five different search algorithms, which include an exhaustive search algorithm (if feasible), a greedy search algorithm as proposed in the previous works, a simple exchange algorithm, a k-exchange algorithm, and our proposed probabilistic subset search (PSS) algorithm.

The exhaustive search guarantees the true optimal design over the discretized domain since it searches over all the $\binom{N}{p}$ possible designs for the one that optimizes the criterion of interest. However, this approach can easily become infeasible, unless N and p are small. The greedy search algorithm is probably the most efficient method in terms of computing time. At each step, this algorithm adds to the current design the next best point that yields the greatest improvement in the optimality criterion. It considers $Np - \frac{p(p-1)}{2}$ cases and is much cheaper than the exhaustive search algorithm. However, by fixing the previously chosen points at each iteration, the greedy search algorithm does not always guarantee the best sampling plan.

The simple exchange algorithm starts with a randomly chosen p-point design. In each iteration, a best point is selected from the remaining $(N - p)$ points and it is added to the current p-point design. Among the $p + 1$ points in the enlarged design, the point that minimizes the reduction of optimality criterion is then removed. This process is repeated until the design obtained in the current iteration is the same as in the previous one. This algorithm is comparably fast, but it can easily be trapped in a local optimum. The k-exchange algorithm is similar to the simple exchange algorithm in the sense that it exchanges points of the current design. This algorithm, however, exchanges $k \ge 1$ points in each iteration; when $k = 1$, the algorithm reduces to the simple exchange algorithm. This feature makes the computation more expensive than the simple exchange algorithm, but reduces the risk of being caught in a local optimum. These exchange algorithms are widely considered in finding optimal experimental designs (Cook et al., 1980; Johnson et al., 1983).

Our proposed PSS algorithm involves the selection of subsets of candidate points based on the probability distributions defined by the eigenvalues and eigenfunctions of the covariance operator of $X(s)$. Specifically, let $\mathcal{X} = \{s_1, \ldots, s_N\}$ be all the time points in the discretized domain, and $\varXi$ be the design space that contains all the p-point designs,

$d_p = \{d_p(1), \ldots, d_p(p)\}$, $d_p(i) \in \mathcal{X}$, $i = 1, \ldots, p$. The algorithm begins with a randomly selected $d_p^0 \in \Xi$. In the kth iteration, we randomly select a subset S_k of l time points from $\mathcal{X} - d_p^{k-1}$ for a given integer l, $1 \le l \le N - p$, where d_p^{k-1} is the design obtained at the $k-1$st iteration. Intuitively, we would want information of the new $X(\cdot)$ at the time points where the between-subject variability is large. When selecting S_k, we thus give higher weight to the points with higher between-subject variability. We consider the following 4 weighting schemes: (i) equal weights for all candidate points, (ii) weights proportional to $\sum_{m=1}^{M} \hat{\rho}_m \hat{\psi}_m^2(s)$, the estimated variance of $X(s)$ using (1), (iii) weights proportional to $\sqrt{\sum_{m=1}^{M} \hat{\rho}_m \hat{\psi}_m^2(s)}$, and (iv) weights proportional to $\sum_{m=1}^{M} \hat{\rho}_m |\hat{\psi}_m(s)|$. Based on our experience, weighting scheme (ii) is the most efficient in terms of the computing time and achieved design efficiency. In particular, (ii) seems to be the "greediest" in the sense that the variability of the weights of different points is the greatest among the four weighting schemes. This might be the reason for (ii) to perform slightly better than the other methods in the case studies that we considered. We thus define the sampling weights, $\Pi = (\pi_{s_1}, \ldots, \pi_{s_N})^T$, of the N candidate points as

$$\pi_{s_i} = \widehat{\mathrm{Var}}(X(s_i)) = \sum_{m=1}^{M} \hat{\rho}_m \hat{\psi}_m^2(s_i), \quad i = 1, \ldots, N.$$

The elements of S_k are randomly selected without replacement from $\mathcal{X} - d_p^{k-1}$ with probability proportional to π_{s_i}. This in turn gives a reduced set of candidate design points; namely, $\mathcal{X}_k = d_p^{k-1} \cup S_k \subset \mathcal{X}$. With $\mathcal{X}_k$, the design space of p-point designs is reduced to $\Xi_k = \{d_p = ((d_p(j)))_{j=1,\ldots,p} : d_p(j) \in \mathcal{X}_k, j = 1, \ldots, p\} \subset \Xi$, and the design maximizing the criterion Φ among the $\binom{p+l}{p}$ designs in Ξ_k is then identified as d_p^k. The algorithm then iterates until a stopping rule is met (e.g., when no significant improvement can be expected).

We note that the PSS algorithm will converge to the true optimum when $p \le l$. While it may not guarantee the optimal solution when $p > l$, our case studies suggest that optimal or near-optimal designs can be efficiently achieved with a small l ($< p$). A pseudo code for this algorithm is described below.

Algorithm Probabilistic subset search algorithm

```
INPUT: Candidate points $\mathcal{X}$; initial design $d_p^0$; subset size $l$
OUTPUT: Optimal design $d_p$
  Set $k \leftarrow 1$
  Calculate $\pi_{s_i}$
  while $d_p^{k-1}$ does not satisfy the stopping rule do
     Choose a subset $S_k$ whose elements are randomly selected without replacement from $\mathcal{X} - d_p^{k-1}$ with probability proportional to $\pi_{s_i}$
     $\mathcal{X}_k = S_k \cup d_p^{k-1}$ and $\Xi_k = \{d_p : d_p(j) \in \mathcal{X}_k, j = 1, ..., p\}$
     Compute $\Phi(\cdot)$ for the $\binom{p+l}{p}$ designs in $\Xi_k$
     Find $d_p^* = \arg\max_{d_p \in \Xi_k} \Phi(d_p)$
     $d_p^k \leftarrow d_p^*$
     $k \leftarrow k + 1$
  end while
```

Instead of considering all the $\binom{N}{p}$ possible designs, our proposed PSS algorithm searches over $\binom{p+l}{p}$ designs in each iteration, thus it can drastically reduce the computational effort in finding optimal designs, especially when l is small. We note that, although a large l may help to improve the achieved design efficiency, the required computing time will also be increased. Our experience suggests that, with a small l (e.g., $l = 3$ or 5), the PSS algorithm can often give satisfactory results (see also Sections 4 and 5); this algorithm is thus expected to save much in computational resources.

In addition, by randomly selecting candidate points to form the subset S_k, we equip the PSS algorithm with the ability to avoid being caught in local optimum. Moreover, we take advantage of the estimated variance of $X(s)$ to determine the selection probability of S_k to help accelerate the convergence of the algorithm. In what follows, we demonstrate the performance of the PSS algorithm and compare it with the other previously mentioned approaches.

4. Simulation studies

In our simulation studies, we assume some known (finite) sets of eigenfunctions for $X(s)$. Three types of eigenfunctions that we consider include: Fourier basis functions, periodic but non-Fourier basis functions, and non-periodic functions. Without loss of generality, we assume that $s \in \mathcal{S} = [0, 1]$ (See S.2 of the supplementary documents). The measurement errors are assumed to be independent and follow a standard normal distribution. Moreover, we assume for simplicity that the mean functions for $X(s)$ and $Y(t)$ are zero for simplicity, although our results still apply for non-zero mean functions.

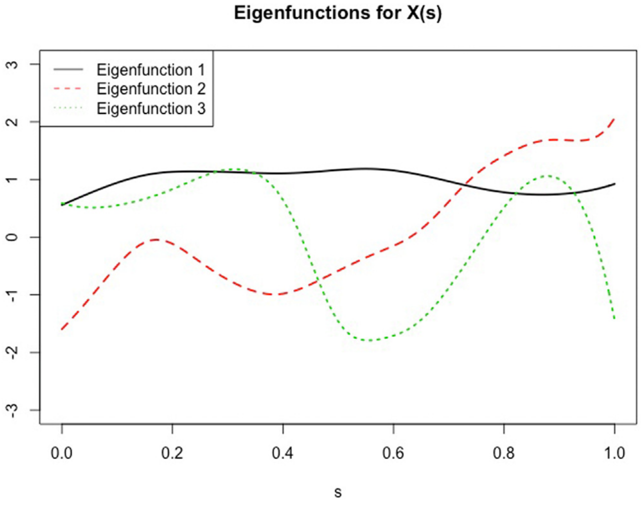

Fig. 1. Eigenfunctions for $X(s)$: The solid (black), broken (red), dashed (green) curves represent the first, second, and third eigenfunctions, respectively.

In the first scenario, we assume that $(M, K) = (3, 3)$, and the first M eigenfunctions of $X(s)$ are

$$\psi_1(s) = C\frac{1}{\sqrt{1+s}}\sin\left(\frac{2\pi}{\ln 2}\ln(1+s)\right),$$
$$\psi_2(s) = C\frac{1}{\sqrt{1+s}}\sin\left(\frac{\pi}{\ln 2}\ln(1+s)\right),\quad\text{and}$$
$$\psi_3(s) = C\frac{1}{\sqrt{1+s}}\sin\left(\frac{4\pi}{\ln 2}\ln(1+s)\right),$$

with eigenvalues $\rho_m = 8, 2, 1$, respectively. Here, C is a normalizing constant, making $\int_S \psi_m^2(s)ds = 1$.

We note that the eigenfunctions for $Y(t)$ and corresponding eigenvalues do not play a role in the search for optimal sampling designs. The B_{KM} matrix, whose (k, m)th element is the b_{km} as defined in (2), is assumed as

$$B_{KM} = \begin{bmatrix} 0.5 & -0.5 & 0.5 \\ 1.5 & -1.0 & 0.5 \\ -0.5 & -0.5 & 0.5 \end{bmatrix}.$$

As the second scenario, we assume that the eigenfunctions of $X(s)$ are

$$\psi_1(s) = 1,$$
$$\psi_2(s) = \sqrt{2}\sin(2\pi s),$$
$$\psi_3(s) = \sqrt{2}\cos(4\pi s),\quad\text{and}$$
$$\psi_4(s) = \sqrt{2}\sin(6\pi s),$$

with eigenvalues $\rho_m = 10, 5, 2, 1$, respectively.

The B_{KM} matrix is set to be

$$B_{KM} = \begin{bmatrix} 3 & -2 & 2 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

For the third scenario, we again assume $M = K = 3$, but the eigenfunctions of $X(s)$ are as in Fig. 1. To demonstrate the applicability of our method, we randomly generate these eigenfunctions.

Eigenvalues for $X(s)$ are assumed to be $\rho_m = 10, 5, 1$, respectively, and

$$B_{KM} = \begin{bmatrix} 0.5 & -1.0 & 0.5 \\ 0.5 & 0.5 & 1.5 \\ 1.5 & 1.0 & 0.5 \end{bmatrix}.$$

With these scenarios, we compare below different search algorithms in terms of their efficiency and effectiveness in optimizing (7).

Table 1
Average of 100 relative efficiencies of achieved designs that maximize the optimality criterion Φ.

	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$
PSS-3	0.8889	0.9431	0.9597	0.9671	0.9717	0.9759
PSS-5	0.8889	0.9435	0.9597	0.9671	0.9717	0.9760
Simple exchange	0.8615	0.9387	0.9570	0.9657	0.9708	0.9751
3-Exchange	0.8807	0.9408	0.9593	0.9668	0.9716	0.9759
Greedy	0.8889	0.9297	0.9512	0.9637	0.9699	0.9743
Exhaustive	0.8889	0.9438	0.9597	0.9671	*infeasible*	

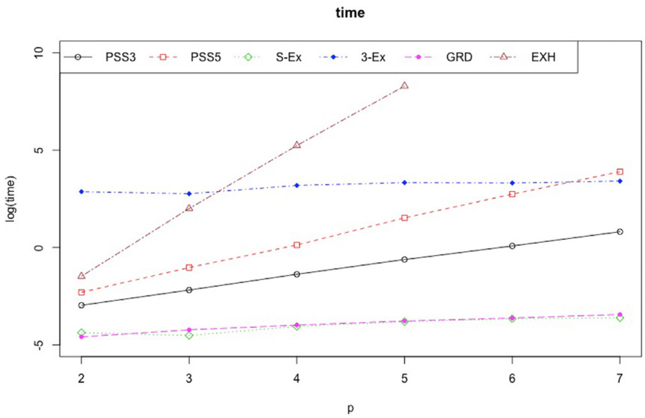

Fig. 2. Log of 100 average computing time in second; S-Ex, 3-Ex, GRD and EXH are for simple exchange, 3-exchange, greedy search and exhaustive search algorithms, respectively.

4.1. Comparison of search algorithms

In the first simulation study, we compare the search algorithms introduced in Section 3 under Scenario 1. The results for Scenarios 2 and 3 are in S.3 and S.4 of the supplementary document, respectively, and they convey similar information. In this study, we discretize the time domain with the grid spacing of 0.01 and consider the criterion $\Phi(\mathbf{s}) = (RE_X(\mathbf{s})+RE_Y(\mathbf{s}))/2$. With this setting, we use the previously described algorithms to find the best p-point designs for $p = 2, 3, \ldots, 7$.

For the simple exchange and k-exchange algorithms, the initial design is chosen randomly, and we also set $k = 3$. For the PSS algorithm, two different sizes (l) of subsets are considered, namely $l = 3$ and 5. The corresponding algorithms are then denoted as PSS-3 and PSS-5, respectively. The search is terminated when the obtained designs remain the same in $L = 30$ consecutive iterations. This number of iterations is decided via some simulation studies. Although a smaller L may reduce computing time, the achieved design efficiency is not as satisfactory. For cases with $L > 30$, we observed that the search requires a long computing time without yielding a significantly improved design efficiency.

For comparison, we report the computing time (in seconds) and the relative design efficiency achieved by these search algorithms. A search is deemed to be infeasible when the required CPU time is longer than 120 min. We note that the average computing time and achieved design efficiency are the corresponding averages obtained by running each algorithm 100 times with different random initial designs. All the simulations are implemented on a desktop computer with a 3.40 GHz Intel i7-2600 8-core processor.

Fig. 2 depicts the logarithm of the required CPU time of each algorithm. The average achieved design efficiencies of these algorithms are presented in Table 1 and Fig. 3. As shown in Fig. 2, the computing time of the exhaustive search algorithm increases exponentially and it quickly becomes infeasible, especially when $p > 5$. The greedy search algorithm is the fastest, but it often generates designs with inferior efficiencies, as shown in Table 1 and Fig. 3. The simple exchange algorithm is as fast as the greedy search algorithm, but as shown in Fig. 3, it can be unstable in terms of the achieved design efficiency. This is because the final design tends to be over-dependent on the initial design. The k-exchange algorithm takes a longer time but is more stable than the simple exchange algorithm. The PSS-3 and PSS-5 algorithms can achieve the highest design efficiencies without requiring much computing time. Following these simulation study results, we recommend the use of the PSS-3 algorithm given its speed and achieved design efficiency.

4.2. Robustness of estimated optimal sampling schedule

Here, we study the robustness of our obtained designs with respect to the variation in the eigen-pairs that are estimated from different simulated data sets. Specifically, we generate the simulation data $U_{il} = X_i(s_{il})+\varepsilon_{il}$ based on the eigenvalues

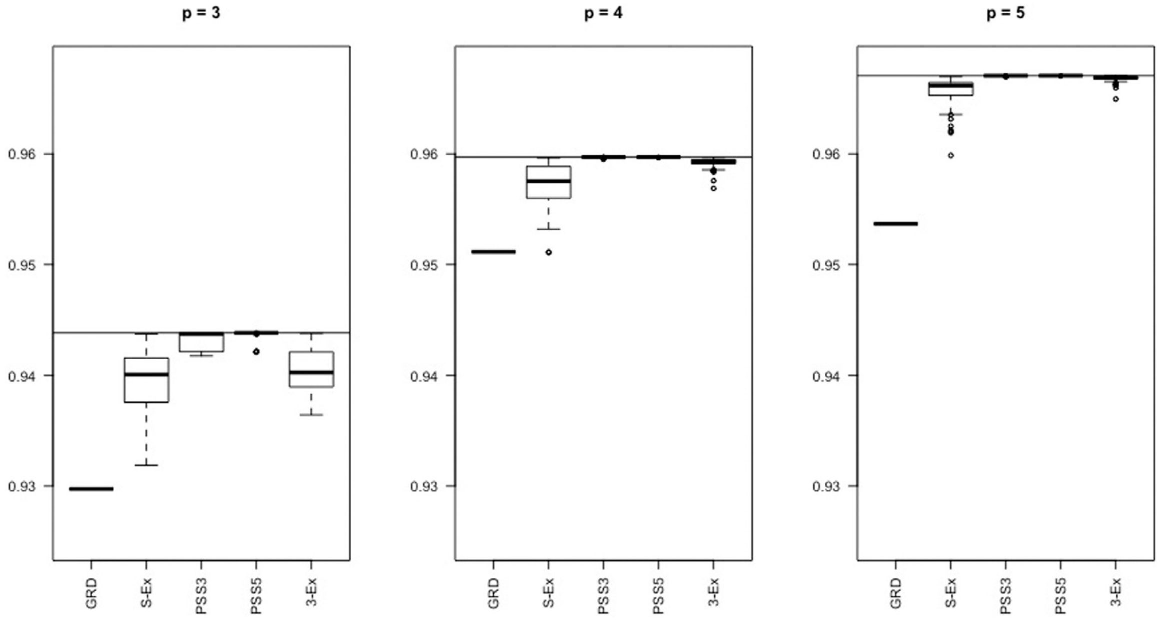

Fig. 3. Boxplots of 100 achieved relative efficiencies. Horizontal lines indicate the theoretical upper-bound obtained from the exhaustive search algorithm.

Table 2
Mean (standard deviation) of 200 *ARE*s obtained by PSS and greedy search algorithms.

		Scenario 1		Scenario 2		Scenario 3	
		PSS	greedy	PSS	greedy	PSS	greedy
$p = 3$	$n = 10$	.0602 (.0675)	.0739 (.0746)	.0834 (.0740)	.0840 (.0752)	.0487 (.0502)	.0534 (.0458)
	$n = 100$	.0082 (.0115)	.0281 (.0182)	.0487 (.0558)	.0845 (.0597)	.0317 (.0288)	.0379 (.0304)
	$n = 1000$	.0021 (.0013)	.0212 (.0095)	.0143 (.0116)	.0447 (.0406)	.0109 (.0114)	.0227 (.0167)
$p = 4$	$n = 10$	.0492 (.0528)	.0494 (.0571)	.0721 (.0701)	.0735 (.0663)	.0366 (.0383)	.0320 (.0363)
	$n = 100$	.0088 (.0061)	.0161 (.0121)	.0299 (.0256)	.0442 (.0395)	.0294 (.0300)	.0223 (.0190)
	$n = 1000$	.0030 (.0030)	.0111 (.0059)	.0079 (.0110)	.0320 (.0163)	.0128 (.0092)	.0134 (.0070)
$p = 5$	$n = 10$	.0422 (.0478)	.0393 (.0510)	.0464 (.0442)	.0397 (.0391)	.0302 (.0341)	.0233 (.0262)
	$n = 100$	.0073 (.0091)	.0095 (.0105)	.0268 (.0213)	.0259 (.0235)	.0257 (.0293)	.0174 (.0147)
	$n = 1000$	.0024 (.0019)	.0047 (.0027)	.0151 (.0103)	.0202 (.0098)	.0088 (.0053)	.0106 (.0063)
$p = 6$	$n = 10$	.0418 (.0520)	.0349 (.0489)	.0391 (.0399)	.0330 (.0414)	.0336 (.0386)	.0243 (.0305)
	$n = 100$	.0059 (.0090)	.0068 (.0055)	.0221 (.0171)	.0171 (.0185)	.0245 (.0272)	.0159 (.0144)
	$n = 1000$	.0015 (.0011)	.0040 (.0022)	.0089 (.0065)	.0076 (.0036)	.0048 (.0034)	.0076 (.0049)

and eigenfunctions specified for Scenarios 1 to 3; $i = 1, \ldots, n$, and $l = 1, \ldots, L_i$.. We consider three different numbers of subjects, namely $n = 10$,100 and 1000. The ith subject has L_i observations where $L_i \sim \text{Uniform}\{2, \ldots, 10\}$, and ε_{il}'s are assumed to be independent and follow normal distribution with variance $\sigma_X = 1$. For each simulated data set, we estimate eigenvalues and eigenfunctions, which in turn give the 'estimated' optimality criterion $\hat{\Phi}$ for $\Phi = [RE_X + RE_Y]/2$. For given number of sampling points $p = 3, 4, 5$ and 6, we find the optimal sampling times $\hat{\mathbf{s}}^*$ maximizing $\hat{\Phi}$ by using the PSS-3 and greedy search algorithms. For comparison purposes, we also define the absolute relative error (ARE) as follows:

$$ARE = \frac{\Phi(\mathbf{s}^*) - \Phi(\hat{\mathbf{s}}^*)}{\Phi(\mathbf{s}^*)}.$$

Here, $\mathbf{s}^*$ is obtained by the respective algorithm to maximizes the 'true' Φ that is obtained from the 'true' eigenvalues and eigenfunctions specified for each scenario. We calculate 200 *ARE*s from 200 independent simulations, and report their mean and standard deviation in Table 2.

From Table 2, it can be seen that the ARE decreases with n, and designs obtained from PSS-3 tend to give a smaller ARE than that of the greedy search, especially when $n = 1000$. When $n = 10$, designs obtained by both algorithms can have an average efficiency loss of about 2.5%-8.5%. Having a good estimate of the eigen-pairs, or a robust design that perform relatively well for a variety of estimated eigen-pairs, can thus be helpful. This is a future research interest.

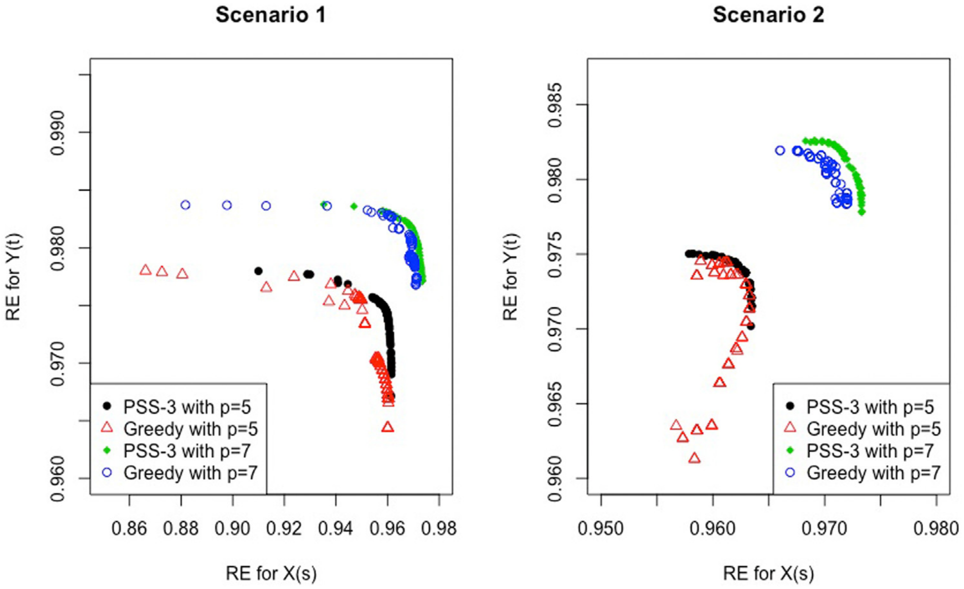

Fig. 4. Optimal designs for different weights generated by PSS-3 and greedy search.

4.3. Weighted sum criteria

In this simulation study, we consider compound optimality criteria of (7). We allow the weights to increase from 0.01 to 0.99 in steps of 0.01, i.e., $W = (0.01, 0.02, \dots, 0.99)$. The greedy search algorithm and the PSS-3 algorithm are used for finding the optimal design for each given weight. For clarity, we focus on the cases with $p = 5$ and 7. Fig. 4 shows the relative efficiencies of the obtained designs for $X(s)$ and $Y(t)$ for Scenarios 1 and 2. Note that the results for Scenario 3 are included in S.5 of the supplementary document, where they provide similar information. Clearly, the PSS algorithm generates better designs than the greedy algorithm. Furthermore, probably due to the characteristics of greedy algorithm, we do not observe a concave Pareto front from its outputs. In contrast, the PSS algorithm gives clear Pareto fronts in Fig. 4. Therefore, we can say that the designs obtained by the PSS-3 algorithm outperform those found by the greedy algorithm. Moreover, it is clear that the relative efficiencies achieved by the designs with $p = 7$ are higher than those with $p = 5$.

Among these obtained designs, there are several ways of choosing one design that fits best the experimenter's needs. First, we may select the design whose relative efficiency has the smallest Euclidean distance to the ideal point $(1, 1)$. For example, for Scenario 1 when $p = 7$, such a design can be obtained from PSS with $w = 0.54$. Another approach is to define a targeted relative efficiency for one criterion and then find the design that maximizes the relative efficiency for the other criterion. For instance, we want to find the best design with the relative efficiency for $Y(t)$ to be at least 0.98. In this case, using the PSS algorithm, we can find that the designs with $w = 0.52$ or 0.53 are the best for maximizing the efficiency for $X(s)$. We note that since the time domain is discretized, the exact same designs may be obtained for different values of w. The relative efficiencies of this particular design in recovering $X(s)$ and predicting $Y(t)$ are 0.9718 and 0.9801, respectively. Moreover, an additional approach is to find the design that makes the two relative efficiencies to be as similar as possible; i.e., to find the design that minimizes $\|RE_X(\mathbf{s}_i^*) - RE_Y(\mathbf{s}_i^*)\|^2$ among the designs on the Pareto front. In this case, $w = 0.99$ is the best weight that satisfies this requirement.

We also study the selection of p by considering the Euclidean distance between the achieved (RE_X, RE_Y) and the ideal point $(1, 1)$. This distance is expected to decrease with p as also observed from Fig. 5. Denoting the minimum Euclidean distance for p sampling design as d_p, a simple method for selecting p is by finding

$$\hat{p} = \min_{p} \left\{ \frac{d_1 - d_p}{d_1} > \delta \right\}, \tag{8}$$

where $0 < \delta < 1$ is a user-specified percentage of improvement from the design with 1 observation. To find $\hat{p}$, we set $\delta = 0.9$ in this simulation study. The appropriate $\hat{p} = 7$ in Scenario 1, while $\hat{p} = 4$ is chosen by the method in (8) for Scenarios 2 and 3. Fig. 5 also suggests that the reduction in the distance between (RE_X, RE_Y) and $(1, 1)$ tends to be smaller as p becomes larger.

4.4. Performance of designs

Here, we compare the performance of the designs obtained from PSS-3 that maximize F_X of (4) and F_Y of (5) separately with random designs and equally spaced designs. For this comparison, we generate a total of 200 'true' paired trajectories

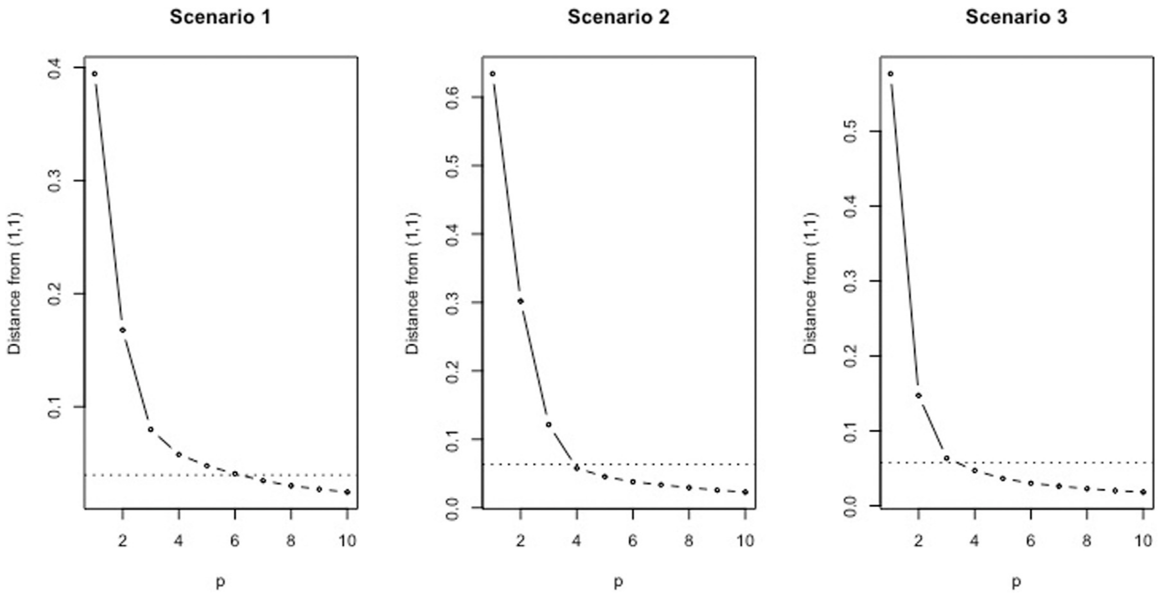

Fig. 5. Minimum distance from (1, 1) for each p with dashed horizontal threshold line for $\delta = 0.9$.

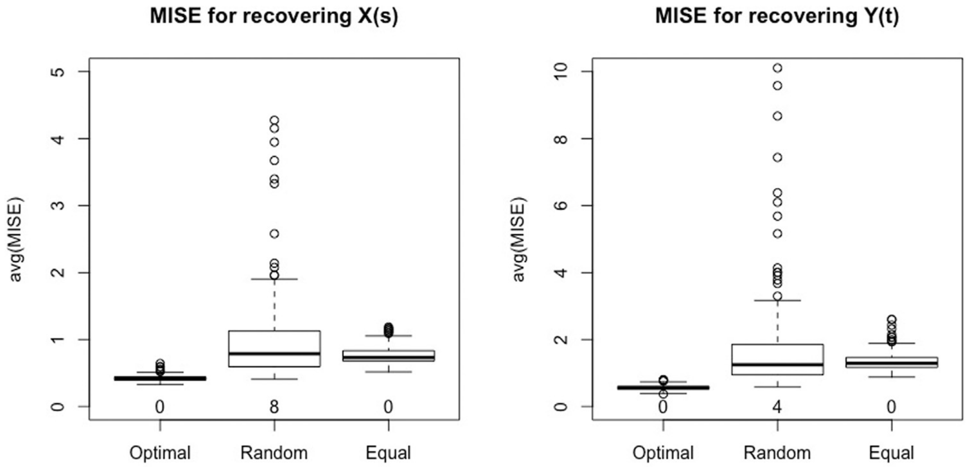

Fig. 6. MISE boxplot for each criterion with $p = 5$ under Scenario 3: bottom numbers imply the number of outlying data over the range.

$(X(\cdot), Y(\cdot))$ according to the assumed eigenfunctions and eigenvalues by using (1) and (2) after replacing the infinite sums with finite sums. For each design that is being compared, we then randomly generate 100 sets of noisy observations U_{il}'s at the p time points in the design, and follow the procedures introduced in Section 2 to give 100 pairs of predicted trajectories for each true $(X(\cdot), Y(\cdot))$. For comparison purposes, the mean integrated squared error (MISE) of the prediction is then calculated for $X(\cdot)$ and $Y(\cdot)$ over these 200 × 100 combinations for each design.

According to Tables 3, 4, and 5, the designs obtained by the PSS algorithm clearly outperform random designs and equally spaced designs at minimizing the MISE; this should be an expected result. Moreover, the MISE decreases as p increases for our obtained designs, but this is not always true for other designs. It is also noteworthy that the decrements get smaller and smaller as we increase the number of measurements, p.

Fig. 6 shows the boxplots of the averages of 100 MISEs (avg(MISE)) generated by these designs for recovering $X(s)$ and predicting $Y(t)$ under Scenario 1 with $p = 5$. As presented there, optimal designs perform better than equally spaced designs or random designs.

5. Applications

We apply our proposed methods to several real data sets, from Alzheimer's disease neuroimaging, primary biliary cirrhosis, and eggs laid by Mediterranean fruit flies. We obtain the single-objective optimal sampling schedules for

Table 3
Mean of 200 MISE under Scenario 1.

		$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
Optimal	X	0.6519	0.5071	0.4213	0.3444	0.2958
	Y	0.9504	0.7001	0.5636	0.4832	0.4128
Random	X	2.2911	1.9220	1.2866	0.9144	0.7322
	Y	3.6951	2.9531	1.8411	1.4708	1.0971
Equal	X	6.4478	2.2956	0.7654	0.6632	0.4734
	Y	12.8957	2.3867	1.3562	1.0576	0.7863

Table 4
Mean of 200 MISE under Scenario 2.

		$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
Optimal	X	1.8197	0.8045	0.6581	0.5496	0.4780
	Y	6.2370	4.6401	3.4349	2.7679	2.3692
Random	X	4.1733	4.3292	2.5480	2.0622	1.3177
	Y	21.0827	20.0287	11.5562	10.6416	6.8009
Equal	X	10.4178	1.8575	2.5727	0.6678	1.4897
	Y	38.3678	6.3009	8.3204	3.2615	4.5380

Table 5
Mean of 200 MISE under Scenario 3.

		$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
Optimal	X	0.7284	0.5308	0.4109	0.3441	0.3000
	Y	1.8296	1.3788	1.0556	0.8310	0.7046
Random	X	2.6253	1.6662	1.1862	0.8063	0.6497
	Y	6.8553	4.2408	3.0290	2.1180	1.6674
Equal	X	1.7843	1.0465	0.6024	0.4615	0.3681
	Y	7.7042	3.5754	1.9516	1.3259	1.0557

Table 6
Optimal sampling schedules for recovering ADAS and HCI curves for the different numbers of measurements.

	$X(s)$: ADAS-Cog ($\widehat{RE}_X$)	$Y(t)$: HCI ($\widehat{RE}_Y$)
$p=2$	75.0, 86.0 (0.680)	80.0, 87.0 (0.817)
$p=3$	68.0, 77.0, 86.5 (0.790)	79.5, 81.5 87.5 (0.864)
$p=4$	64.0, 68.0, 80.0, 87.0 (0.826)	67.5, 80.0, 80.5, 87.5 (0.900)
$p=5$	63.0, 68.5, 75.5, 84.5, 87.5 (0.860)	67.5, 80.0, 80.5, 81.0, 87.5 (0.919)
$p=6$	63.0, 67.0, 69.5, 77.0, 84.5, 87.5 (0.882)	68.0, 79.5, 80.0, 80.5, 85.5, 88.0 (0.931)
$p=7$	63.0, 65.5, 68.0, 70.0, 79.5, 85.0, 88.0 (0.893)	68.0, 79.5, 80.0, 80.5, 81.0, 86.0, 88.0 (0.940)

functional predictor $X(s)$ and functional/scalar response $Y(t)/Y$ that maximize (4) and (5), respectively, and the bi-objective optimal sampling schedules that maximize (7).

5.1. Alzheimer's disease neuroimaging data

Alzheimer's disease (AD) is one of the most common forms of dementia. The Alzheimer's Disease Assessment Scale-Cognitive Subscale (ADAS-Cog) is designed to assess the level of cognitive impairment in AD. It is a clinical rating and consists of 11 tasks measuring the disturbances in memory, language, praxis, attention, and other cognitive abilities which are often referred to as the core symptoms of AD (Kolibas et al., 2000). On the other hand, another assessment score, the hypometabolic convergence index (HCI), is taken to reflect in a single measurement the extent to which the pattern and magnitude of cerebral hypometabolism in an individual's fluorodeoxyglucose positron emission tomography (FDG-PET) image correspond to ADAS-Cog in a probable AD patient, and it is generated by using a fully automated voxel-based image-analysis algorithm (Chen et al., 2011). In this study, we use ADAS-Cog as an explanatory function and HCI as a response function. The source data come from the ADNI database, the data sheet "148-n3355-FDG-ALL-Info", and the HCI scores are calculated by the neuroimaging lab in the Banner Alzheimer Institute.

We use the data from patients between 63 and 88 years of age, which covers most of the measurements in the data set. There are 269 patients who have observations within this age range, and the number of measurements varies from 1 to 8 with a median of 2. A spaghetti plot of the data is shown in Fig. 7. This data set is used to estimate the unknown quantities needed in calculating the optimality criteria using the methods introduced in Section 2. We then find the optimal sampling schedule for recovering predictor and response functions with the grid spacing of 0.5 year.

The single-objective optimal p-point designs generated by our proposed PSS-3 algorithm with $p = 2, \dots, 7$ are shown in Table 6. The achieved $\widehat{RE}_X$ and $\widehat{RE}_Y$ are also reported there. For example, the optimal sampling points for ADAS-Cog

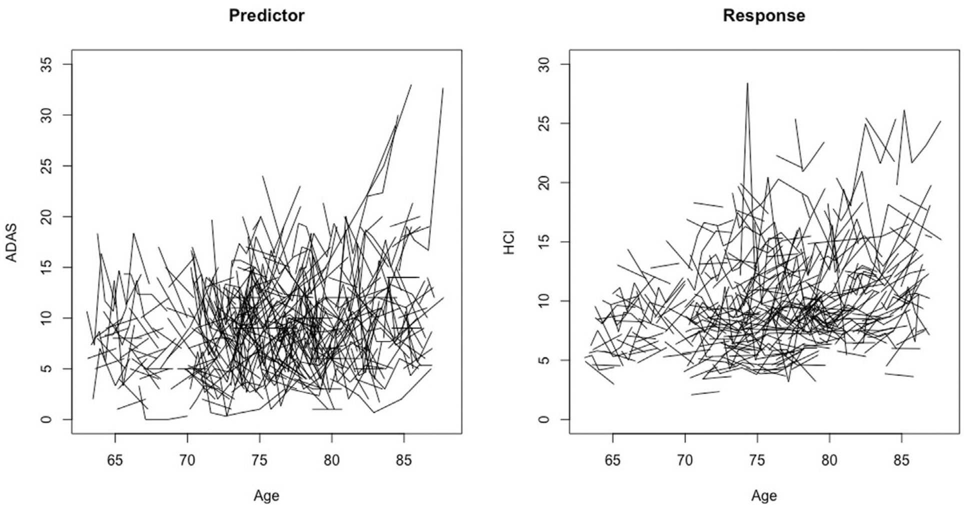

Fig. 7. Left figure: Alzheimer's disease assessment scale (ADAS-Cog); Right figure: hypometabolic convergence index.

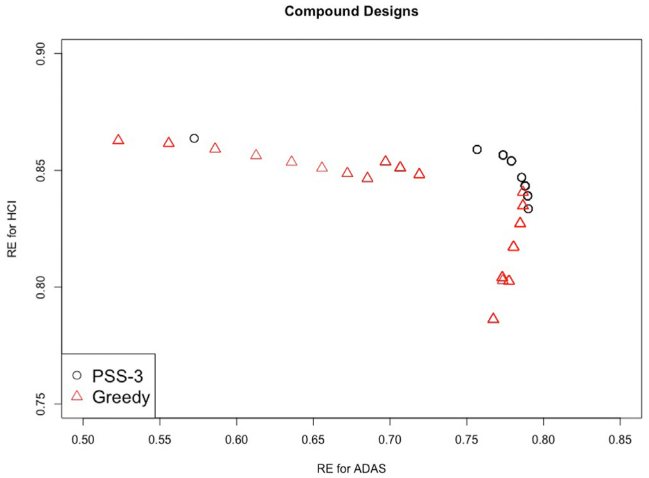

Fig. 8. Performance of compound optimal designs for different weights with $p = 3$. Black circles are designs generated by the PSS algorithm, and red triangles are designs generated by the greedy search algorithm.

when $p = 2$ are 75.0 and 86.0 with $\widehat{RE}_X = 0.68$. We note that, for a finite p, the optimal sampling schedule for the predictor function $X(\cdot)$ is not necessarily the best sampling point for the response function $Y(\cdot)$; see also Park et al. (2018) for a similar observation. This is mainly because we have different optimality criteria, namely (4) and (5), for $X(\cdot)$ and $Y(\cdot)$. In particular, the criterion for $Y(\cdot)$ depends not only on the criterion for $X(\cdot)$, but also on the correlation of $X(\cdot)$ and $Y(\cdot)$ through $\hat{B}_{KM}$. Unlike the greedy search algorithm, these points are not part of the optimal design for $p = 3$. We also observe that the optimal sampling time points for $X(\cdot)$ tend to be spread out over the age range, whereas those for $Y(\cdot)$ occur at the later time points where the variability seems larger (see Fig. 7).

In Fig. 8, we present our obtained compound designs for different weights when $p = 3$. The designs obtained by the greedy search algorithm are also provided, and clearly, the PSS-3 algorithm generates better designs than greedy search algorithm. We can choose weights $w = 0.52, \ldots, 0.60$ to find the design that minimizes the Euclidean distance between the relative design efficiencies and $(1, 1)$. We note that the same designs are obtained from several different weights w due to a large grid spacing for discretizing the time domain. This also suggests that, for large grid spacings, we may reduce the size of W (i.e. the number of different weights) when obtaining compound designs for approximating the Pareto front. We also consider the choice of the appropriate number of sampling schedules defined in (8). According to Fig. 9, when $\hat{p} = 7$, the Euclidean distance from $(1, 1)$ becomes less than $(1 - \delta)d_1$ where $\delta = 0.9$.

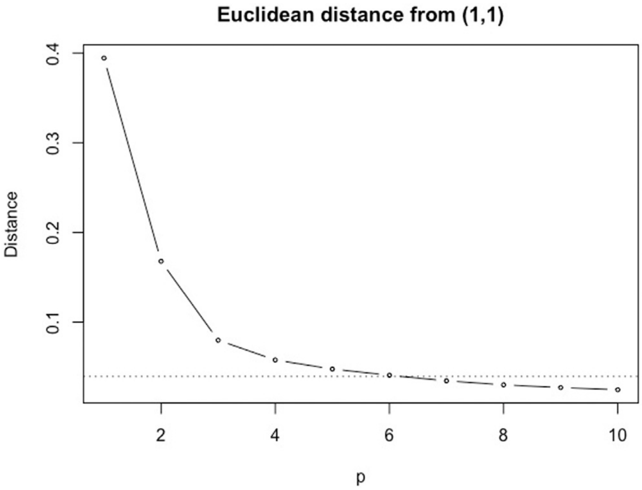

Fig. 9. Minimum distance from (1, 1) for each p with dashed horizontal threshold line.

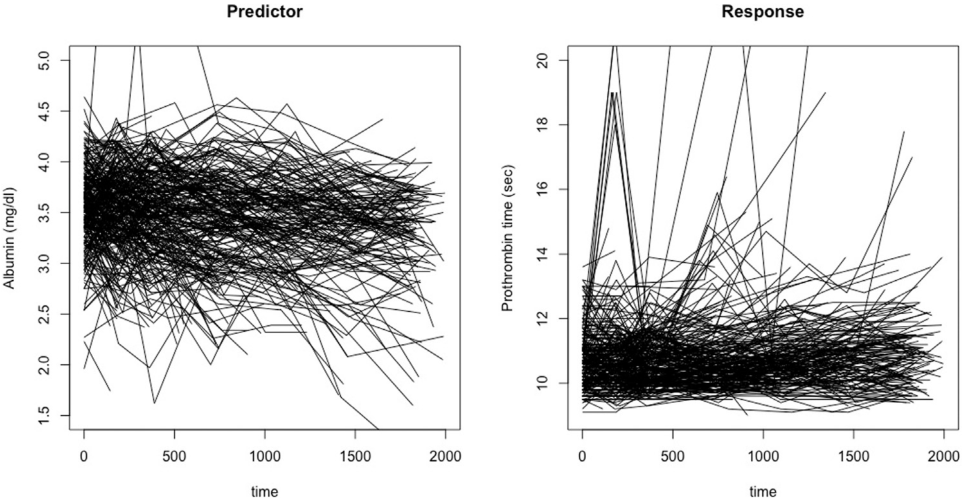

Fig. 10. Left figure: Albumin Level; Right figure: Prothrombin time.

5.2. Primary biliary cirrhosis data

Primary biliary cirrhosis (PBC) is a fatal immune disorder which slowly and gradually destroys biliary ducts of the liver (Dancygier, 2010). Damaged biliary ducts result in the damage in liver over time, and it is called cholestasis. It is known to occur more commonly in women with the ratio 9:1 female to male with a prevalence of one per 3–4000 people (Mülhaupt et al., 2006). The data were collected at the Mayo Clinic from 1975 to 1984. There were 312 PBC patients that participated in the randomized trial, and they were supposed to visit at 6 months, 1 year, and annually thereafter. However, the data were collected on a sparse and irregular grid, since many of patients missed some of scheduled visits, and this results in a different number of measurements for each patient.

Yao et al. (2005b) discussed the relationship between albumin in mg/dl as a predictor function and prothrombin time in seconds as a response function. To find the optimal designs, we use the data from 2000 days of 312 PBC patients, shown in Fig. 10. The number of visits varied from 1 to 8 with the median of 5. Optimal designs for recovering the predictor function (albumin), and for predicting the response function (prothrombin time) are obtained by the proposed method for the different numbers of measurements; compound designs with different weights are also generated. We consider 51 candidate days from day 0 to day 2000 in steps of 40 days.

Optimal sampling schedules for the predictor and response functions with different number of measurements are shown in Table 7. Similar to the previous application, the more observations we make, the higher the relative efficiencies obtained. As shown in Fig. 11, compound designs generated by the PSS algorithm outperform those obtained from the greedy search algorithm.

Table 7
Optimal sampling schedules for recovering albumin level and prothrombin time curves for the different numbers of measurements.

	$X(s)$: Albumin level ($\widehat{RE}_X$)	$Y(t)$: Prothrombin time ($\widehat{RE}_Y$)
$p = 2$	680, 1480 (0.784)	960, 1360 (0.725)
$p = 3$	600, 1080, 1600 (0.831)	0, 920, 1320 (0.772)
$p = 4$	40, 760, 1160, 1600 (0.861)	0, 840, 1200, 1400 (0.818)
$p = 5$	40, 760, 1080, 1560, 1600 (0.882)	0, 800, 1120, 1200, 1560 (0.843)
$p = 6$	0, 720, 760, 1160, 1560, 1600 (0.900)	0, 800, 960, 1160, 1240, 1560 (0.859)
$p = 7$	0, 720, 760, 1120, 1200, 1600, 1640 (0.909)	0, 40, 800, 960, 1160, 1200, 1600 (0.872)

Table 8
Optimal sampling schedules for recovering the curve of the number of eggs laid on a particular day and the remaining total number of eggs for the different numbers of measurements.

	$X(s)$: Number of eggs ($\widehat{RE}_X$)	$Y(t)$: Remaining eggs ($\widehat{RE}_Y$)
$p = 2$	10, 17 (0.623)	7, 25 (0.672)
$p = 3$	8, 14, 19 (0.702)	8, 24, 25 (0.739)
$p = 4$	7, 10, 16, 21 (0.753)	7, 8, 24, 25 (0.760)
$p = 5$	7, 10, 14, 16, 21 (0.781)	7, 8, 18, 24, 25 (0.770)
$p = 6$	7, 8, 11, 14, 19, 22 (0.804)	7, 8, 13, 18, 24, 25 (0.783)
$p = 7$	7, 8, 9, 12, 16, 17, 24 (0.823)	7, 8, 12, 18, 23, 24, 25 (0.794)

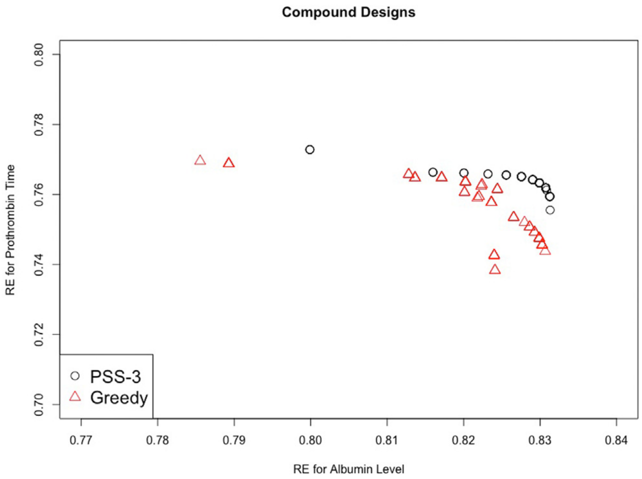

Fig. 11. Performance of compound optimal designs for different weights. Black circles are designs generated by the PSS algorithm, and red triangles are designs generated by the greedy search algorithm.

5.3. *Eggs laid from mediterranean fruit flies data*

One advantage of the PSS algorithm is its versatility in different situations including finding optimal sampling plans for predicting a scalar response. To demonstrate its applicability, we find optimal sampling schedules for recovering trajectory of the number of eggs laid daily over 25 days (predictor function) and for predicting the total number of eggs remaining during lifetime (scalar response). Ji and Müller (2017) found the optimal sampling times for recovering trajectory and scalar response, and used the exhaustive search algorithm and greedy search algorithm to obtain optimal designs. Their FOptDes function of the 'fdapace' package in R is available for finding such designs using either the exhaustive search or greedy search algorithms. The observations were made every day, i.e., the number of eggs was measured 25 times per subject.

We also obtain the optimal design using PSS-3 for recovering the predictor function and for predicting scalar response with the same grid spacing for the time domain as in Ji and Müller (2017). Our obtained optimal designs are shown in Table 8. We compare the computing time of PSS-3 and the FOptDes function of 'fdapace' package in R for $p = 2, 3, 4, 5, 6,$ and 7. Table 9 shows the average of 100 computing time for PSS-3 and the FOptDes function with the exhaustive search. The optimal sampling schedules for $Y(t)$ obtained by PSS-3 are identical to those generated by the FOptDes function, which guarantees the optimal design. However, PSS-3 mostly, but not always, generates consistent designs compared to those generated by the FOptDes function. It is because the FOptDes function uses the approximated values to compute optimality criteria for $X(s)$, while PSS-3 uses the exact values for computation. As p increases, the computing time increases

Table 9
Average (standard deviation) of 100 computing times (in seconds) for finding optimal designs.

	$X(s)$: Predictor		Y: Response	
	PSS-3	FOptDes	PSS-3	FoptDes
$p = 2$	$0.0325(5.43 \times 10^{-3})$	$0.1272(7.90 \times 10^{-3})$	$0.0310(1.49 \times 10^{-2})$	$0.1971(7.38 \times 10^{-3})$
$p = 3$	$0.0670(1.54 \times 10^{-2})$	$0.2547(7.99 \times 10^{-3})$	$0.0616(1.70 \times 10^{-2})$	$0.2998(3.14 \times 10^{-2})$
$p = 4$	$0.1465(2.65 \times 10^{-2})$	$0.9459(4.41 \times 10^{-2})$	$0.1305(1.89 \times 10^{-2})$	$0.8473(6.88 \times 10^{-2})$
$p = 5$	$0.2923(4.80 \times 10^{-2})$	$3.5728(4.51 \times 10^{-2})$	$0.2595(4.24 \times 10^{-2})$	$2.8424(6.57 \times 10^{-2})$
$p = 6$	$0.5535(7.10 \times 10^{-2})$	$12.147(2.23 \times 10^{-1})$	$0.5198(7.31 \times 10^{-2})$	$9.4726(4.49 \times 10^{-1})$
$p = 7$	$0.9896(1.16 \times 10^{-1})$	$33.156(1.94 \times 10^{-1})$	$0.9238(5.92 \times 10^{-2})$	$25.238(2.62 \times 10^{-1})$

for both algorithms. However, the PSS-3 algorithm outperforms the FOptDes in terms of computing time, especially when p is large.

6. Discussion

We are concerned with the optimal design for both recovering the predictor function and predicting the response function under a functional linear regression model. We present a framework that makes it possible to efficiently find the optimal sampling schedule that improves precision in functional trajectory recovery.

In our simulation studies, the PSS algorithm is reasonably fast and generates good single-objective and bi-objective compound designs. We also show that our obtained optimal designs outperform and result in significant improvement over random designs or equally spaced designs in terms of minimizing MISE. For future subjects with a limited number of measurements, it is better to allocate observations based on the proposed designs in this paper. Moreover, the PSS algorithm can generate the Pareto front with different weights for compound optimality criteria. Among these designs, one can choose the design that satisfies the experimenter's requirements. Here, we focus on cases where $p \leq 7$, and our proposed PSS with $l = 3$ is observed to work well. One may increase l for an improved design efficiency. But in our study, this improvement does not seem to compensate for the increased computing time. As for an increased p (> 7), the PSS-3 and greedy search attain very similar design efficiencies, but the latter algorithm is faster than the former, and thus should be considered.

The approach presented herein is useful and widely applicable to real problems, but it has some limitations. First, to apply this method, we must have some information about the unknown quantities in the functional regression model. One possible way to handle this difficulty is by estimating these quantities from the existing data. Moreover, if the estimates are not as accurate as we desire, then the designs obtained by our method will not be very useful; that is, the quality of estimates is crucial for effective experimental designs. Thus, one future research interest is to find suitable experimental plans to provide good model parameter estimations. It should also be useful to obtain designs under a model misspecification of, e.g., the eigen-pairs of the covariance operator of the predictor function. Another future research direction is to extend the simple functional linear regression to the multiple functional linear regression. Moreover, in this paper, we focus on the optimal sampling plans for the case where only the observations of predictor function are to be collected. In some situations, we will be able to make observations on the response axis as well, although the number of observations may be very limited. The predictor and response functions are connected through the regression coefficient surface, namely $\beta(s, t)$. One may consider using the observations of a new subject to update the estimated coefficient surface $\hat{\beta}(s, t)$. The quality of the updated $\hat{\beta}(s, t)$ depends on the sampling plans for the predictor function and for the response function. Finding optimal sampling plans in both domains is thus important and an extension of our proposed method to such a situation is a future research interest.

Acknowledgments

We thank the anonymous referees for raising questions that resulted in an improvement of this paper.

Funding

This work was supported by the National Science Foundation, USA [grant number CMMI-17-26445, DMS-13-52213]; and the National Center for Theoretical Sciences in Taiwan.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2020.106925.

References

Castro, P.E., Lawton, W.H., Sylvestre, E.A., 1986. Principal modes of variation for processes with continuous sample curves. Technomatrics 28 (4), 329–337.

Chen, K., Ayutyanont, N., Langbaum, J.B., Fleisher, A.S., Reschke, C., Lee, W., Liu, X., Bandy, D., Alexander, G.E., Thompson, P.M., Shaw, L., Trojanowski, J.Q., Jr., Jack, R., C., Landau, S.M., Foster, N.L., Harvey, D.J., Weiner, M.W., Koeppe, R.A., Jagust, W.J., Reiman, E.M., 2011. Characterizing Alzheimer's disease using a hypometabolic convergence index. NeuroImage 56, 52–60.

Cook, R.D., Nachtsheim, C.J., 1980. A comparison of algorithms for constructing exact d-optimal designs. Technomatrics 22 (3), 315–324.

Dai, X., Hadjipantelis, P.Z., Han, K., Ji, H., 2018. fdapace: Functional data analysis and empirical dynamics. https://github.com/functionaldata/tPACE, R package version 0.4.0.

Dancygier, H., 2010. Clinical Hepatology: Principles and Practice of Hepatobiliary Diseases. Springer.

Ferraty, F., 2014. Regression on functional data: methodological approach with application to near-infrared spectrometry. J. Soc. Francaise Stat. 155 (2), 100–120.

He, G., Müller, H.G., Wang, J.L., 2000. Extending correlation and regression from multivariate to functional data. Asymptot. Stat. Prob. 1–14.

Hsing, T., Eubank, R., 2015. Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators. Wiley.

James, G.M., Hastie, T.J., Sugar, C.A., 2000. Principal component models for sparse functional data. Biometrika 87 (3), 587–602.

Ji, H., Müller, H.G., 2017. Optimal designs for longitudinal and functional data. J. R. Stat. Soc. Ser. B Stat. Methodol. 79 (3), 859–876.

Jiang, C.R., Aston, J.D., Wang, J.L., 2009. Smoothing dynamic positron emission tomography time courses using functional principal components. NeuroImage 47, 184–193.

Johnson, M.E., Nachtsheim, C.J., 1983. Some guidelines for constructing exact d-optimal designs on convex design space. Technometrics 25 (3), 271–277.

Kolibas, E., Korinkova, V., Novotny, V., Vajdickova, K., D., H., 2000. ADAS-cog (Alzheimer's Disease Assessment Scale-cognitive subscale)–validation of the Slovak version. Eur. Psychiatry 15 (2), 443.

Lai, T.L., Shih, M.C., Wong, S.P., 2006. A new approach to modeling covariate effects and individualization in population. J. Pharmacokinet. Pharmacodyn. 33 (1), 49–74.

Li, C., 2017. Statistical Methods for Functional and Complex Data (Doctoral dissertation). North Carolina State University, Raleigh, NC.

Mülhaupt, B., Smith, A.D., 2006. Medical Care of the Liver Transplant Patient: Total Pre-, Intra- and Post-Operative Managment. Wiley-Blackwell.

Pan, R., Saleh, M., 2019. On designing experiments for a dynamic response modeled by regression splines. Appl. Stoch. Models Bus. Ind. 1–17.

Park, S.Y., Xiao, L., Willbur, J.D., Staicu, A.M., Jumbe, N.L., 2018. A joint design for functional data with application to scheduling ultrasound scans. Comput. Statist. Data Anal. 122, 101–114.

Peng, J., Paul, D., 2009. A geometric approach to maximum likelihood estimation of the functional principal components from sparse longitudinal data. J. Comput. Graph. Statist. 18 (4), 995–1015.

Ramsey, J.O., Silverman, B.W., 2005. Functional Data Analysis. Springer.

Rice, J.A., 2004. Functional and longitudinal data analysis: perspectives on smoothing. Statist. Sinica 14, 631–647.

Rice, J.A., Silverman, B.W., 1991. Estimating the mean and covariance structure nonparametrically when the data are curves. J. R. Stat. Soc. Ser. B Stat. Methodol. 53 (1), 233–243.

Saleh, M., Kao, M.-H., Pan, R., 2017. Design d-optimal event-related functional magnetic resonance imaging experiments. J. R. Stat. Soc. Ser. C. Appl. Stat. 66 (1), 73–91.

Sood, A., James, G.M., Tellis, G.J., 2009. Functional regression: a new model for predicting market penetration of new products. Mark. Sci. 28 (1), 36–51.

Wu, M., Diez-Roux, A., Raghunathan, T.E., Sanchez, B.N., 2018. FPCA-based method to select optimal sampling schedules that capture between-subject variability in longitudinal studies. Biometrics 74 (1), 229–238.

Xiao, L., Li, C., Checkley, W., Crainiceanu, C., 2018. Fast covariance estimation for sparse functional data. Stat. Comput. 28, 511–522.

Yao, F., Müller, H.G., Wang, J.L., 2005a. Functional data analysis for sparse longitudinal data. J. Amer. Statist. Assoc. 100 (470), 577–590.

Yao, F., Müller, H.G., Wang, J.L., 2005b. Functional linear regression analysis for longitudinal data. Ann. Statist. 33 (6), 2873–2903.

Yao, F., Müller, H.G., Clifford, A.J., Dueker, S.R., Follett, J., Lin, Y., Buchholz, B.A., Vogel, J.S., 2003. Shrinkage estimation for functional principal component scores with application to the population kinetics of plasma folate. Biometrics 59, 676–685.

Zhu, H., Versace, F., Cinciripini, P.M., Rausch, P., Morris, J.S., 2018. Robust and Gaussian spatial functional regression models for analysis of event-related potentials. NeuroImage 181, 501–512.