Statistics and Computing (2022) 32:30
https://doi.org/10.1007/s11222-021-10066-y

Sparse functional partial least squares regression with a locally sparse slope function

Tianyu Guan[1] · Zhenhua Lin[2] · Kevin Groves[3] · Jiguo Cao[4]

Received: 5 October 2020 / Accepted: 5 November 2021 / Published online: 9 March 2022

Abstract

The partial least squares approach has been particularly successful in spectrometric prediction in chemometrics. By treating the spectral data as realizations of a stochastic process, the functional partial least squares can be applied. Motivated by the spectral data collected from oriented strand board furnish, we propose a sparse version of the functional partial least squares regression. The proposed method aims at achieving locally sparse (i.e., zero on certain sub-regions) estimates for the functional partial least squares bases, and more importantly, the locally sparse estimate for the slope function. The new approach applies a functional regularization technique to each iteration step of the functional partial least squares and implements a computational method that identifies nonzero sub-regions on which the slope function is estimated. We illustrate the proposed method with simulation studies and two applications on the oriented strand board furnish data and the particulate matter emissions data.

Keywords Partial least squares · B-spline basis functions · Functional data analysis · Functional linear regression · Locally sparse · Principal components

1 Introduction

In this article, we develop a sparse version of the functional partial least squares. Our proposed method provides locally sparse estimators for the functional partial least squares basis functions. More importantly, the proposed method is able to produce a locally sparse estimate for the slope function in the functional linear regression model.

Our work is motivated by an oriented strand board (OSB) furnish research conducted by FPInnovations, a private not-for-profit organization that specializes in the creation of solutions in support of the Canadian forest sector's global competitiveness. In this study, a novel laboratory spectroscopy technique was developed for determining species identification of OSB strands and the relative proportions of sound wood, rot, and bark in OSB fines samples. The log section samples were first retrieved from Canadian OSB mills and then separated into sound wood, rot, and bark groups. A laboratory disk strander was used to prepare strands and a laboratory grinder ground the samples into coarse powder. Vis–NIR (visible and near-infrared) spectroscopy measurement techniques were applied to acquire the spectra traces of the samples with 2150 individual wavelengths ranging from 350 to 2500 nm. Figure 1a displays the spectra traces of 182 OSB fines samples.

Currently, mills do not monitor the key raw material constituents (sound wood, rot, and bark) that play a major role in production operating efficiency and final product attributes. Periodically monitoring raw material can help mills identify problems associated with rot in logs, debarking inefficiency, and species variability. Measurements can also provide data to assist in process adjustments and production planning and budgeting. Figure 1b illustrates spectra traces of OSB fines samples with different proportions of compositions of sound wood. It can be observed in Fig. 1b that samples with different proportions of sound wood have distinct spectra traces and thereby we focus on predicting the proportions of sound wood in OSB fines samples from their spectra traces.

✉ Jiguo Cao
jiguo_cao@sfu.ca

[1] Department of Mathematics and Statistics, Brock University, St. Catharines, ON L2S 3A1, Canada

[2] Department of Statistics and Data Science, National University of Singapore, Singapore, Singapore

[3] Engineered Wood Products Manufacturing, FPInnovations, Vancouver, BC V6T 1Z4, Canada

[4] Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, BC V5A1S6, Canada

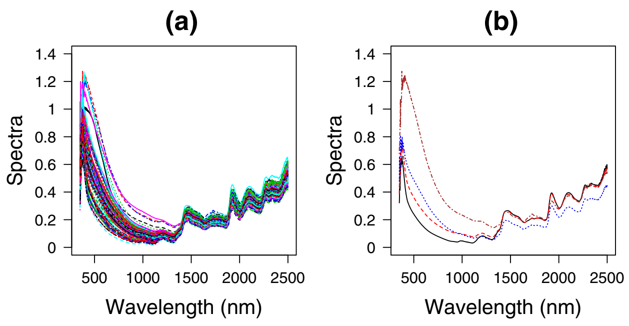

Fig. 1 **a** Spectra traces of 182 OSB fines samples. **b** Selected spectra traces of OSB fines samples with different proportions of sound wood (———, 100%; - - - - -, 80%; ······, 40%; -·-·-·- 0%)

Partial least squares (PLS) regression searches for components from the perspective of prediction and takes the response into account, and is often used in spectrometric prediction in chemometrics. Cook and Forzani (2019) pointed out that the best asymptotic behavior of PLS is reached in abundant regressions in which many predictors contribute information about the response, which occurs in chemometrics applications. For instance, in the OSB furnish study, the spectra traces were acquired at 2150 individual wavelengths from 350 to 2500 nm. On the one hand, we can treat the spectra as 2150 separate variables and apply the PLS method. However, this approach ignores the continuity, smoothness, and order of the 2150 measurements. For this reason, instead of considering them as separate variables, we treat the spectra traces from 2150 individual wavelengths as realizations of a stochastic process. Therefore we aim to develop a functional partial least squares method. At the same time, we are also interested in detecting the sub-ranges of the wavelengths of the spectra that have no significant contribution on the prediction of the proportions of sound wood. The literature on PLS is abundant; see Wold (1975), Helland (1990), and Garthwaite (1994), for example. The PLS first gained popularity in chemometrics (e.g., Frank and Friedman 1993; Martens and Næs 1992), and more recently has been applied to other fields, such as bioinformatics (Boulesteix and Strimmer 2006) and image detection (Schwartz et al. 2009). The PLS method was also extended to process functional data by Preda et al. (2007) and Delaigle and Hall (2012a) for classification. Escabias et al. (2007) adapted PLS to the functional logit model. Delaigle and Hall (2012b) developed an alternative formulation of PLS in the functional setting that led to detailed theoretical results. Delaigle and Hall (2012b) also pointed out that compared with the functional principal components analysis (Lin et al. 2016; Sang et al. 2017; Nie et al. 2018; Nie and Cao 2020; Shi et al. 2021), the functional partial least squares (FPLS) captures the interaction between the response and the functional predictor using a smaller number of components, which leads to more parsimonious and interpretable bases. In addition, PLS was employed in functional linear models by Preda and Saporta (2005) to determine the slope function. Reiss and Ogden (2007) proposed a smooth estimator of the slope function by combining the smoothing method and the PLS in the functional linear regression model. Although these endeavors can produce satisfactory estimates for the slope function using PLS, they do not specifically focus on the locally sparse estimation.

To address the aforementioned problem, we develop a new method, which we call *sparse functional partial least squares method* (SFPLS), that simultaneously produces locally sparse estimates for the functional partial least squares basis functions and the slope function. Achieving locally sparse estimates for the FPLS basis functions is not difficult. However, the locally sparse estimates for the FPLS basis functions do not directly lead to a locally sparse estimate for the slope function, because the estimated sparse sub-regions of each basis function may not overlap. Our proposed SFPLS method combines the ideas of the dimension reduction via PLS, the B-spline expansion, and a functional regularization technique which is called functional smoothly clipped absolute deviation (fSCAD) penalty (Lin et al. 2017). Similar to PLS, the SFPLS method is an iterative procedure. In each iteration, we obtain a locally sparse FPLS basis by employing the fSCAD penalty and identify the nonzero sub-regions, called active regions. On the subspace of the active regions, we update all the FPLS basis functions and estimate the slope function. This idea is motivated by Chun and Keleş (2010), who proposed a sparse partial least squares (SPLS) method for simultaneous dimension reduction and variable selection.

Our method is related to Chun and Keleş (2010). However, the two proposals differ in several ways. Chun and Keleş (2010) used the PLS method in linear regression models. By imposing sparsity in the midst of the dimension reduction step, they achieved simultaneous dimension reduction and variable selection. We extend their method to the context of functional linear regression models. Chun and Keleş (2010) applied an L_1-penalty to promote sparsity, while we employ an fSCAD penalty to achieve the local sparseness of the functional partial least squares bases and the slope function. In addition, we impose a roughness penalty to regularize the smoothness of the estimators.

We highlight our contributions in this article as follows. First, we develop a sparse version of the functional partial least squares, which is easy to interpret. The nonzero subregions of the functional partial least squares bases represent the active regions on which the response has major correlations with the functional covariate. Second, compared with the FPLS method and the functional principal component method, the proposed SFPLS method provides a more accurate estimator of the locally sparse slope function and achieves a smaller prediction error using fewer components. Third, the proposed method provides a locally sparse estimate for the slope function using FPLS bases. There are some related works on the locally sparse modeling of the slope function. James et al. (2009) pioneered the locally sparse estimation of the slope function. They expanded the slope function using a simple grid basis for the empirical results. Zhou et al. (2013) and Lin et al. (2017) studied the locally sparse slope function by expressing it as a linear combination of B-spline basis functions. Hall and Hooker (2016) and Guan et al. (2020) investigated the truncated functional linear regression models. To the best of our knowledge, no work has been done on using the FPLS bases to determine the locally sparse slope function due to the difficulty of deriving the FPLS bases that have overlapping zero sub-regions. Preda and Saporta (2005) and Reiss and Ogden (2007) estimated the slope function using PLS. However, their primary interests are not the interpretability, and their methods do not allow locally sparse modeling. Finally, our proposed method is computationally efficient. This is partly achieved by a closed-form solution for the estimator of the coefficients for expanding each sparse FPLS basis function.

We structure the rest of the paper as follows. In Sect. 2, we introduce the proposed SFPLS method and provide its computational details. In Sect. 3, simulation results are presented to numerically evaluate the performance of the proposed method. Applications of the SFPLS approach to the OSB furnish data and the particulate matter emissions data are given in Sect. 4. Section 5 concludes the paper. Additional discussions are provided in an online supplementary document.

2 Methodology

2.1 The first sparse functional partial least squares basis function

We consider the first FPLS basis function here and discuss the estimation of the other basis functions in a separate subsection. Suppose we observe data pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$, which are independently and identically distributed as (X, Y), where X is a random function defined on an interval $[0, T]$ and Y is a random scalar generated by $Y = \mu + \int_0^T X(t)\beta(t)\mathrm{d}t + \varepsilon$, with μ being the intercept, $\beta(t)$ the slope function, and ε representing the noise that is independent of X. The first FPLS basis function might be found by

$$\max_{w} \quad \mathrm{Cov}^2\left(Y, \int_0^T X(t)w(t)\mathrm{d}t\right), \qquad \text{subjectto} \quad \|w\|_2^2 = 1. \tag{1}$$

To estimate $w(t)$, we utilize B-spline basis functions that are detailed in de Boor (2001). Let $\boldsymbol{B}(t) = (B_1(t), \ldots, B_{M+d}(t))^{\mathrm{T}}$ be a vector that contains $M + d$ B-spline basis functions. Each basis function is defined on $[0, T]$ with degree d and $M + 1$ equally spaced knots $0 = t_0 < t_1 < \cdots < t_M = T$, which is a piecewise polynomial of degree d. B-spline basis functions are well known for their compact support property, i.e., each basis function is positive over at most $d + 1$ adjacent subintervals. We approximate $w(t)$ by $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{B}(t)$ with $\boldsymbol{b} = (b_1, \ldots, b_{M+d})^{\mathrm{T}}$ representing the coefficients for expanding the FPLS basis function $w(t)$. Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^{\mathrm{T}}$ and denote by $\boldsymbol{U}$ the $n \times (M + d)$ matrix with elements $u_{ij} = \int_0^T X_i(t)B_j(t)\mathrm{d}t$. If we denote $\boldsymbol{V}_0$ the $(M + d) \times (M + d)$ matrix with the elements $v_{0ij} = \int_0^T B_i(t)B_j(t)\mathrm{d}t$, then $\boldsymbol{b}$ can be estimated by

$$\max_{\boldsymbol{b}} \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{b}$$
$$\text{subject to} \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{b} = 1,$$

where $\boldsymbol{H} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{U}$.

To derive the first sparse FPLS basis function, we introduce a surrogate $r(t)$ of $w(t)$ and impose the fSCAD penalty on the surrogate $r(t)$. We approximate the surrogate function $r(t)$ by $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}(t)$, where $\boldsymbol{c} = (c_1, \ldots, c_{M+d})^{\mathrm{T}}$. For $m > 0$, let $\boldsymbol{B}^{(m)}(t) = \left(B_1^{(m)}(t), \ldots, B_{M+d}^{(m)}(t)\right)^{\mathrm{T}}$ denote the vector of the m-th derivatives of the B-spline basis functions. The first sparse FPLS basis function is then obtained by

$$\hat{w}(t) = \frac{\hat{\boldsymbol{c}}^{\mathrm{T}}}{\|\hat{\boldsymbol{c}}\|_2}\boldsymbol{B}(t), \tag{2}$$

where $\hat{\boldsymbol{c}} = (\hat{c}_1, \ldots, \hat{c}_{M+d})^{\mathrm{T}}$ solves

$$\begin{aligned} \min_{\boldsymbol{b},\boldsymbol{c}} \quad & -\kappa\boldsymbol{b}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{b} + (1-\kappa)(\boldsymbol{b}-\boldsymbol{c})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{b}-\boldsymbol{c}) \\ & + \gamma\left\|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}^{(m)}\right\|_2^2 + \frac{M}{T}\int_0^T p_\lambda\left(\left|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}(t)\right|\right)\mathrm{d}t + \delta\left\|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}\right\|_2^2, \\ \text{subjectto} \quad & \boldsymbol{b}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{b} = 1, \end{aligned} \tag{3}$$

with nonnegative parameters κ, γ, λ and δ. In this formulation, $p_\lambda(\cdot)$ is a SCAD penalty function proposed by Fan and Li (2001), which is defined on $[0, \infty]$ as

$$p_\lambda(u) = \begin{cases} \lambda u & \text{if } 0 \le u \le \lambda \\ -\frac{u^2-2a\lambda u+\lambda^2}{2(a-1)} & \text{if } \lambda < u < a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } u \ge a\lambda, \end{cases}$$

where a is a constant suggested to be 3.7. In (2), we use the rescaled $\hat{\boldsymbol{c}}$ as the B-spline coefficients, which has a norm 1.

In the objective function, the first term measures the interaction between the response and the latent component, while the second term enforces the surrogate function to be close to the FPLS basis function in the same context. The third term is a roughness penalty that controls the smoothness of the estimated surrogate function. In practice, we choose $m = 2$, which corresponds to measuring the roughness of a function by its integrated curvature. The fourth term is the fSCAD penalty which is designed to regularize the sparseness of the estimated surrogate function. The last term in the objective function aims to stabilize the estimation procedure.

Let $\boldsymbol{V}$ denote the $(M+d)\times(M+d)$ matrix with the elements $v_{ij} = \int_0^T B_i^{(m)}(t)B_j^{(m)}(t)\mathrm{d}t$. The roughness penalty term yields $\gamma\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{c}$. We follow Lin et al. (2017) to approximate the fSCAD penalty by the local quadratic approximation. Let $\boldsymbol{V}_{0j} = \int_{t_{j-1}}^{t_j}\boldsymbol{B}(t)\boldsymbol{B}^{\mathrm{T}}(t)\mathrm{d}t$ and define

$$G_\lambda\left(r^{(0)}\right) = \sum_{l=1}^{M} p_\lambda\left(\frac{\|r^{(0)}_{[l]}\|_2}{\sqrt{T/M}}\right) - \frac{1}{2}\sum_{l=1}^{M} p'_\lambda\left(\frac{\|r^{(0)}_{[l]}\|_2}{\sqrt{T/M}}\right)\frac{\|r^{(0)}_{[l]}\|_2}{\sqrt{T/M}},$$

$$\boldsymbol{Q}^{(0)}_\lambda = \frac{1}{2}\sum_{l=1}^{M}\left(\frac{p'_\lambda\left(\|r^{(0)}_{[l]}\|_2\sqrt{M/T}\right)}{\|r^{(0)}_{[l]}\|_2\sqrt{T/M}}\boldsymbol{V}_{0l}\right),$$

where $\|r^{(0)}_{[l]}\|_2 = \left(\int_{t_{l-1}}^{t_l}\left|r^{(0)}(t)\right|^2\mathrm{d}t\right)^{1/2}$. Then

$$\frac{M}{T}\int_0^T p_\lambda\left(\left|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}\right|\right)\mathrm{d}t \approx \boldsymbol{c}^{\mathrm{T}}\boldsymbol{Q}^{(0)}_\lambda\boldsymbol{c} + G_\lambda\left(r^{(0)}\right).$$

A choice for $r^{(0)}$ is the first FPLS basis function obtained by solving (1). Since $G_\lambda\left(r^{(0)}\right)$ does not depend on $\boldsymbol{b}$ and $\boldsymbol{c}$, we remove it from the objective function. We may write the last term in the objective function as $\delta\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{c}$. Therefore the objective function in (3) can be approximated by

$$-\kappa\boldsymbol{b}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{b} + (1-\kappa)(\boldsymbol{b}-\boldsymbol{c})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{b}-\boldsymbol{c}) + \gamma\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{c} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{Q}^{(0)}_\lambda\boldsymbol{c} + \delta\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{c}. \quad (4)$$

2.2 Computational method

To solve (3), we iterate between solving for $\boldsymbol{b}$ for fixed $\boldsymbol{c}$ and solving for $\boldsymbol{c}$ for fixed $\boldsymbol{b}$. Estimating $\boldsymbol{b}$ given $\boldsymbol{c}$ becomes the following problem

$$\min_{\boldsymbol{b}}\ -\kappa\boldsymbol{b}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{b} + (1-\kappa)(\boldsymbol{b}-\boldsymbol{c})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{b}-\boldsymbol{c}),$$
$$\text{subjectto } \boldsymbol{b}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{b} = 1.$$

Since $\boldsymbol{V}_0$ is symmetric and positive definite, we can write $\boldsymbol{V}_0 = \boldsymbol{W}_0\boldsymbol{W}_0$, where $\boldsymbol{W}_0$ is symmetric. Let $\boldsymbol{b}_* = \boldsymbol{W}_0\boldsymbol{b}$ and $\boldsymbol{c}_* = \boldsymbol{W}_0\boldsymbol{c}$. Define $\boldsymbol{Z} = \boldsymbol{W}_0^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y}$. For $0 < \kappa < 1/2$, $\boldsymbol{b}$ can be obtained by

$$\boldsymbol{W}_0^{-1}\arg\min_{\boldsymbol{b}_*}\ \left(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{b}_* - \kappa'\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{c}_*\right)^{\mathrm{T}}\left(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{b}_* - \kappa'\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{c}_*\right),$$
$$\text{subjectto } \boldsymbol{b}_*^{\mathrm{T}}\boldsymbol{b}_* = 1, \quad (5)$$

where $\kappa' = (1-\kappa)/(1-2\kappa)$. Now if we fix $\boldsymbol{b}$ in (4), $\boldsymbol{c}$ can be estimated by minimizing

$$(1-\kappa)(\boldsymbol{b}-\boldsymbol{c})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{b}-\boldsymbol{c}) + \gamma\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{c} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{Q}^{(0)}_\lambda\boldsymbol{c} + \delta\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_0\boldsymbol{c}.$$

Based on Theorem 3 of Chun and Keleş (2010), we have Theorem 1 which gives the closed-form solution to minimizing (4) for $0 < \kappa \leq 1/2$.

Theorem 1 *For* $0 < \kappa \leq 1/2$, *the solution to minimizing* (4) *is* $\hat{\boldsymbol{b}} = \boldsymbol{W}_0^{-1}\boldsymbol{Z}/\|\boldsymbol{Z}\|_2$ *and*

$$\hat{\boldsymbol{c}} = (1-\kappa)\left((1-\kappa)\boldsymbol{H} + \gamma\boldsymbol{V} + \boldsymbol{Q}^{(0)}_\lambda + \delta\boldsymbol{V}_0\right)^{-1}\boldsymbol{H}\hat{\boldsymbol{b}}. \quad (6)$$

The proof is relegated to the supplementary document. Based on the solution to minimizing (4), the solution to (3) can be found by iteratively computing (6) until $\hat{\boldsymbol{c}}$ converges. The initial estimate $\hat{\boldsymbol{c}}^{(0)}$ is the first FPLS weight vector (see the supplementary document). Then given $\hat{\boldsymbol{c}}^{(i)}$, we compute $r^{(i)}(t) = \boldsymbol{B}(t)^{\mathrm{T}}\hat{\boldsymbol{c}}^{(i)}$, $\boldsymbol{Q}^{(i)}_\lambda$, and $\hat{\boldsymbol{c}}^{(i+1)} = (1-\kappa)\left((1-\kappa)\boldsymbol{H} + \gamma\boldsymbol{V} + \boldsymbol{Q}^{(i)}_\lambda + \delta\boldsymbol{V}_0\right)^{-1}\boldsymbol{H}\hat{\boldsymbol{b}}$. As pointed out by Lin et al. (2017), $\hat{\boldsymbol{c}}$ found by the local quadratic approximation is a local minimizer. Once $\hat{\boldsymbol{c}}$ is produced, the estimate for the first empirical sparse FPLS basis function is given in (2).

2.3 An algorithm for the sparse functional partial least squares

Incorporating the formulation of the first sparse FPLS basis into the iterative conventional FPLS algorithm (see the supplementary document) enables us to obtain the subsequent sparse FPLS basis functions. However, the sparse FPLS basis functions derived this way are unlikely to have the same zero sub-regions and therefore the estimated slope function might not be sparse. To see this, without loss of generality, we assume $X(\cdot)$ and Y are centered, i.e., $\mathbf{E}X(t) \equiv 0$ and $\mathbf{E}Y \equiv 0$, so that μ is zero. Assume $w_1(t), \ldots, w_K(t)$ are the first K sparse FPLS basis functions. We express the slope function in terms of the sparse FPLS bases, which can be written as $\beta(t) = \sum_{k=1}^{K}\alpha_k w_k(t)$. Let $\boldsymbol{S}$ be the $n \times K$ matrix with elements $s_{ij} = \int_0^T X_i(t)w_j(t)\mathrm{d}t$. The columns of $\boldsymbol{S}$ represent the scores. The coefficient vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^{\mathrm{T}}$ can be estimated using ordinary least squares by $\hat{\boldsymbol{\alpha}} = (\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S})^{-1}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{Y}$. It is obvious that if the basis functions $w_1(t), \ldots, w_K(t)$ do not have overlapping zero sub-regions, then $\hat{\beta}(t) = \sum_{k=1}^{K} \hat{\alpha}_k w_k(t)$ is not a locally sparse estimator.

To address the above problem, we propose the following algorithm. At each iteration, we first produce a sparse FPLS basis function by using (2) and solving (3), based on which we define active regions $\mathcal{A}$, and then we update all the FPLS basis functions on $\mathcal{A}$. The slope function is expanded by the FPLS basis functions and is not zero only on the active regions $\mathcal{A}$. Define $X_{\mathcal{A}}(t) = X(t)I_{(t\in\mathcal{A})}$, where $I_{(\cdot)}$ denotes the indicator function, and let $\boldsymbol{X}_{\mathcal{A}} = \{X_{1\mathcal{A}}, \ldots, X_{n\mathcal{A}}\}$. We first set the initial estimate $\hat{\beta}^{(0)}(t) = 0$ for all $0 \le t \le T$ and set $\mathcal{A}^{(0)} = \varnothing$. Set $\boldsymbol{Y}^{(1)} = \boldsymbol{Y}$. The algorithm is outlined below. For $k = 1, \ldots, K$,

1. Find $\hat{w}(t)$ by using (2) and solving (3) with $\boldsymbol{H} = \left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y}^{(k)}\right)\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y}^{(k)}\right)^{\mathrm{T}}$.
2. Update $\mathcal{A}^{(k)}$ as $\{t : \hat{w}(t) \neq 0\} \cup \mathcal{A}^{(k-1)}$.
3. Update the first k FPLS basis functions with $\boldsymbol{Y}$ and $\boldsymbol{X}_{\mathcal{A}^{(k)}}$ using the conventional FPLS method detailed in the supplementary document.
4. Update $\hat{\beta}^{(k)}(t)$ by using the estimated FPLS basis functions obtained in step 3
5. Update $\boldsymbol{Y}^{(k)} = \left(Y_1^{(k)}, \ldots, Y_n^{(k)}\right)^{\mathrm{T}}$ with $Y_i^{(k)} = Y_i - \int_0^T X_i(t)\hat{\beta}^{(k)}(t)\mathrm{d}t$.

The nonzero active regions are updated at each iteration. For $k = K$, $\mathcal{A}^{(K)}$ is a combination of the nonzero sub-regions for the first K sparse FPLS basis functions. After completion of steps 1 to 5 for $k = 1, \ldots, K$, the estimated SFPLS basis functions are obtained in step 3 when $k = K$, the estimated slope function $\hat{\beta}^{\mathrm{SFPLS}}(t) = \hat{\beta}^{(K)}(t)$, and the estimated nonzero sub-regions for the slope function is $\mathcal{A}^{(K)}$. The main computational complexity of the algorithm at the kth iteration is $O((M+d)^3 + kn^2 + kn(M+d))$. Steps 1, 3, and 5 take most of the computation time of the algorithm. The main computational complexity in step 1 is $O((M+d)^3)$, and steps 3 and 5 add additional complexities $O(kn^2)$ and $O(kn(M+d))$, respectively.

Our algorithm extends the SPLS-NIPALS algorithm proposed by Chun and Keleş (2010) to the functional data context. The SPLS-NIPALS algorithm achieves simultaneous dimension reduction and variable selection by incorporating sparsity in the nonlinear iterative partial least squares (NIPALS) algorithm, a well-known algorithm for PLS (Wold 1975). Chun and Keleş (2010) used an L_1 penalty for sparsity. In comparison, we apply the fSCAD penalty in the algorithm to produce locally sparse estimate for the functional partial least squares basis functions and for the slope function.

In our fitting procedure, there are a few tuning parameters including κ, γ, λ, δ, K, and the parameters for constructing the B-spline basis functions such as the degree d of the B-spline basis and the number of knots $M + 1$. The crucial tuning parameters are the shrinkage parameter λ, the number of latent components K, and the smoothing parameter γ.

We need to determine the number of knots in both steps 1 and 3. In step 1, the cubic B-splines are used to express the SFPLS bases. Cardot et al. (2003) suggested that the choice of M is not important due to the roughness penalty and suggested a relatively large value for M to capture the local features of the functions; see also Marx and Eilers (1999) and Lin et al. (2017). To find K SFPLS components, we need to repeat steps 1-5 of the SFPLS algorithm for K times. To reduce the computational complexity, in step 3 we do not tune the number of knots, instead, we use the number of knots selected by the FPLS method with $\boldsymbol{Y}$ and $\boldsymbol{X}$.

The degree d, which is of less importance, is fixed to a reasonable value, i.e., $d = 3$. As discussed in Theorem 1, we choose $0 < \kappa \le 0.5$. We follow Chun and Keleş (2010) and set $\kappa = 0.1$ in all numerical studies. We discuss the effect of κ in the supplementary document. We set δ to a sensible value to stabilize the estimation procedure. Readers are referred to the supplementary document for a discussion on the effects of κ and δ.

The roughness penalty in (3) regularizes the smoothness of the surrogate function, the zero sub-regions of which determine the active regions $\mathcal{A}$. In other words, if the roughness parameter γ is very small, the resulting active regions will be very irregular and scattered; see the supplementary document for more details. The above algorithm implies that the sparse estimation of the slope function depends highly on the shrinkage parameter λ and the number of components K. Observe that $\mathcal{A}^{(1)} \subseteq \cdots \subseteq \mathcal{A}^{(K)} \subseteq [0, T]$, a large number of components may not lead to a sparse estimation of $\beta(t)$. To encourage model sparsity, the parameters γ, λ and K are tuned by minimizing BIC. We follow Krämer and Sugiyama (2011) to calculate the degree of freedom of the model and BIC.

We should also mention that more in-depth study on the asymptotic properties of the estimator is needed and will be left as our future research work.

3 Simulation studies

In this section, we numerically illustrate the performance of the proposed method via simulations. We first show that compared with the functional principal component regression (FPCR) method and the conventional FPLS method, the proposed SFPLS method provides a more accurate estimator for the slope function and achieves a smaller prediction error using fewer bases. Then we illustrate the proposed method identifies the null regions with great accuracy.

The functional predictors $X_i(t)$ are generated by $X_i(t) = \sum a_{ij} B_j(t)$, where $B_j(t)$ are cubic B-spline basis functions defined on 50 equally spaced knots over [0, 1], and the coefficients a_{ij} are sampled from the standard normal distribution. We consider the following scenarios for $\beta(t)$:

Scenario I: $\beta(t) = I_{(0 \le t < 0.5)}$. $\beta(t)$ is a discontinuous function with a flat nonzero part on [0, 0.5). The zero sub-region is [0.5, 1].

Scenario II: $\beta(t) = \sin(2\pi t) I_{(0 \le t < 0.5)}$. $\beta(t)$ is a continuous function with a trigonometric nonzero part on (0, 0.5). The zero sub-region is [0.5, 1].

Scenario III: $\beta(t) = I_{(0.25 \le t < 0.75)}$. $\beta(t)$ is a discontinuous function with a flat nonzero part on [0.25, 0.75). The zero sub-regions [0, 0.25) and [0.75, 1] are disconnected.

Scenario IV: $\beta(t) = 3t + e^{t^2}\cos(3\pi t) + 1$. $\beta(t)$ is a continuous function without zero sub-regions.

The function $\beta(t)$ in these scenarios are plotted in Figs. 8 and 9. The errors ε_i are independently generated from normal distributions so that the signal-to-noise ratio equals 5. In each study, we run the simulation independently for 100 times with sample sizes $n = 100$ and $n = 500$. For each simulation replicate we also generate a separate test dataset with a sample size of 5000.

First, we estimate $\beta(t)$ from the training dataset using the first K terms of the bases of the SFPLS method, the conventional FPLS method, and the FPCR method, for each of $K = 1, \ldots, 5$. The conventional FPLS method and the FPCR method are detailed in the supplementary document. Cubic B-splines with 101 equally spaced knots are used by the SFPLS method to express the SFPLS bases (step 1 in the SFPLS algorithm). The FPLS method adopts cubic B-splines with the number of knots selected by Cross-Validation (CV) and BIC. To reduce the computational complexity, in step 3 of the SFPLS algorithm, we use the number of knots selected by the FPLS method.

For the proposed SFPLS method, we set $\kappa = 0.1$ and $\delta = 0.1$ in all numerical studies. For each of $K = 1, \ldots, 5$, the smoothing parameter γ and sparse parameter λ are chosen by BIC. We select the tuning parameters of the FPLS and FPCR methods by CV and BIC. For both FPLS and FPCR, the results based on BIC are better than CV. Therefore, for both methods, we provide results based on BIC.

The performance of the estimated $\hat{\beta}(t)$ is evaluated by the integrated squared errors on the zero sub-regions and the nonzero sub-regions, which are, respectively, defined as

$$\mathrm{ISE}_0 = \frac{1}{l_0}\int_{\mathcal{I}_0}\left(\hat{\beta}(t) - \beta(t)\right)^2 \mathrm{d}t$$

$$\text{and} \quad \mathrm{ISE}_1 = \frac{1}{l_1}\int_{\mathcal{I}_1}\left(\hat{\beta}(t) - \beta(t)\right)^2 \mathrm{d}t,$$

where $\mathcal{I}_0$ denotes the zero sub-regions of $\beta(t)$, $\mathcal{I}_1$ denotes the nonzero sub-regions of $\beta(t)$, and l_0 and l_1 are the lengths of the zero and nonzero sub-regions, respectively. For example, in Scenario I, $\mathcal{I}_0 = [0.5, 1]$, $\mathcal{I}_1 = [0, 0.5)$, and $l_0 = l_1 = 0.5$.

The performance of prediction is assessed by prediction mean squared errors (PMSE) on a test dataset that is independent of training data, which is computed by

$$\mathrm{PMSE} = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}}\left(Y_i - \int_0^T X_i(t)\hat{\beta}(t)\mathrm{d}t\right)^2,$$

where n_{test} denotes the sample size of the test dataset, which is 5000 in our simulation settings.

Figure 2 displays boxplots of the integrated squared error, ISE_0, defined on the zero sub-regions for locally sparse $\beta(t)$ in scenarios I, II, and III using the first K, $K = 1, \ldots, 5$, terms of the bases of SFPLS, FPLS, and FPCR for $n = 100$. The results in Fig. 2 suggest that the proposed SFPLS method requires only one component to achieve a smaller ISE_0 than those obtained by the conventional FPLS method and FPCR method using up to five components. In Fig. 3, we present the boxplots of the logarithmic integrated squared error, ISE_1, defined on the nonzero sub-regions for all scenarios for $\beta(t)$ for $n = 100$. It is observed that in general, SFPLS and FPLS perform similarly and they require fewer components than FPCR to reach the same level of ISE_1. Figure 4 shows the boxplots of the logarithmic prediction mean squared errors on the test data based on 100 simulation replications for $n = 100$. We observe that to reach the same level of PMSE, SFPLS needs the fewest terms whereas FPCR requires the most terms. In addition, from both Figs. 3 and 4, we can observe that for Scenario IV where there are no zero sub-regions, the SFPLS method and FPLS method provide

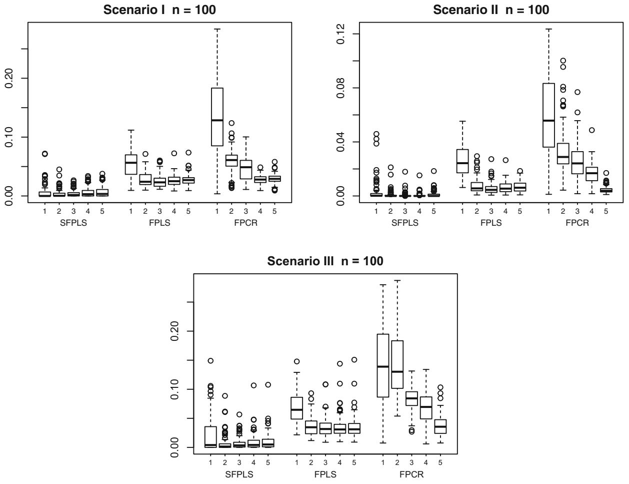

Fig. 2 Boxplots of the integrated squared error, ISE_0, defined on the zero sub-regions for $\hat{\beta}(t)$ using the first K, $K = 1, \ldots, 5$, terms of the bases of SFPLS, FPLS, and FPCR based on 100 simulation replications with $n = 100$. In each subfigure, from left to right, each group of 5 boxplots corresponds to $K = 1$ to $K = 5$

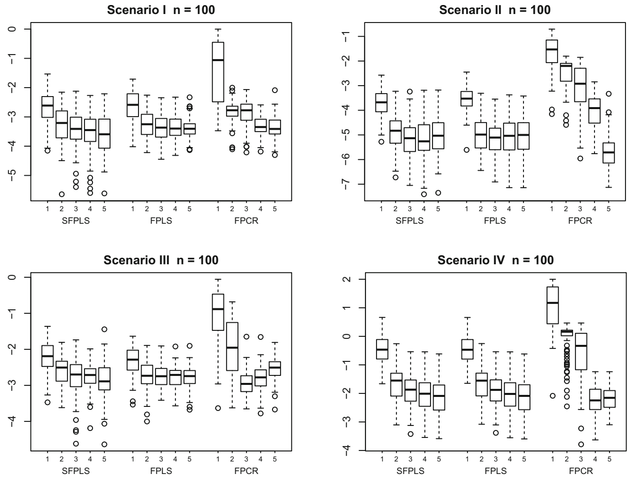

Fig. 3 Boxplots of the logarithmic integrated squared error, ISE_1, defined on the nonzero sub-regions for $\hat{\beta}(t)$ using the first K, $K = 1, \ldots, 5$, terms of the bases of SFPLS, FPLS, and FPCR based on 100 simulation replications with $n = 100$. In each subfigure, from left to right, each group of 5 boxplots corresponds to $K = 1$ to $K = 5$

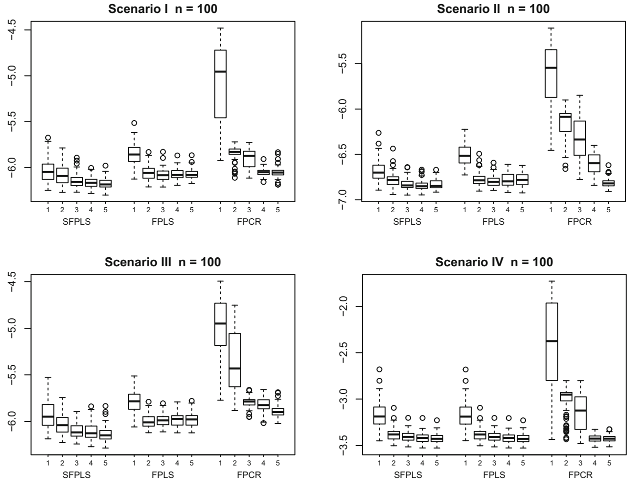

Fig. 4 Boxplots of the logarithmic prediction mean squared errors, PMSE, using the first K, $K = 1, \ldots, 5$, terms of the bases of SFPLS, FPLS, and FPCR on the test data based on 100 simulation replications with $n = 100$. In each subfigure, from left to right, each group of 5 boxplots corresponds to $K = 1$ to $K = 5$

nearly identical results. The results regarding the sample size $n = 500$ are reported in the supplementary document, which are similar to those obtained based on $n = 100$.

To further evaluate the numerical performance of the proposed method, instead of fitting the training dataset using each of the first 1 to 5 components, respectively, we choose the number of components by the approach introduced in Sect. 2.3. We also compare the proposed method with the conventional FPLS method, the regularized version of the functional partial least squares (FPLS_R) proposed by Reiss and Ogden (2007), the FPCR method, and the smooth and locally sparse (SLoS) method proposed by Lin et al. (2017), which is based on locally sparse modeling. The FPLS_R method adopts B-spline basis expansion and incorporates a roughness penalty in the regression. The roughness tuning parameter of FPLS_R is selected by fitting a linear mixed model through restricted maximum likelihood (REML) estimation (Reiss and Ogden 2007). The SLoS method proposed by Lin et al. (2017) achieves a locally sparse estimator of the slope function by applying a novel fSCAD regularization technique.

We use cubic B-splines with 101 equally spaced knots in step 1 of the SFPLS algorithm, and for the FPLS_R and SLoS methods. For the proposed SFPLS method, the smoothing parameter γ, the sparse parameter λ and the number of components K are chosen by the approach presented in Sect. 2.3. In the supplementary document, we discuss the effects of κ, γ, and δ. The FPLS method adopts cubic B-splines with the number of knots and the number of components selected by Cross-Validation (CV) and BIC. The tuning parameters of the FPCR method are selected by CV and BIC. For the FPLS method and FPCR method, the results based on BIC are better than CV. Therefore, we include the BIC results for both methods. For the FPLS_R method, as suggested by Reiss and Ogden (2007), the number of components is chosen by CV, while the roughness parameter is selected by REML. The SLoS estimators are implemented and tuned according to Lin et al. (2017).

We conclude the ISE_0, logarithm of ISE_1, and logarithm of PMSE results for $n = 100$ in Figs. 5, 6, and 7. The results for $n = 500$ are included in the supplementary document. Figure 5 summarizes the ISE_0 of the estimators. The results indicate that the proposed SFPLS method outperforms all the other methods on the zero sub-regions, thanks to the sparse penalty. Figure 6 presents a summary of the logarithm

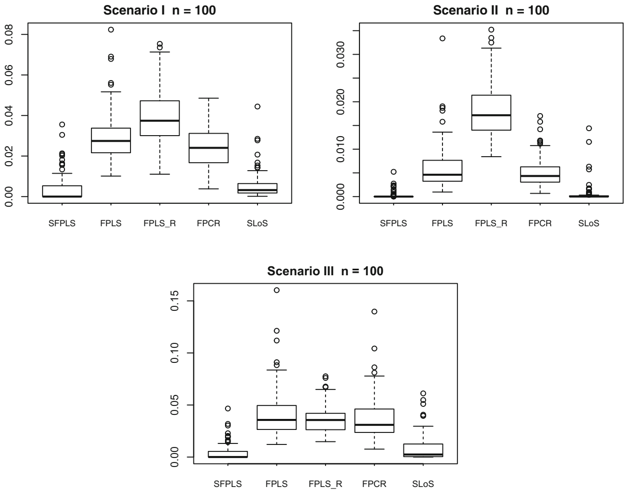

Fig. 5 Boxplots of the integrated squared error, ISE_0, defined on the zero sub-regions for $\hat{\beta}(t)$ based on 100 simulation replications with $n = 100$

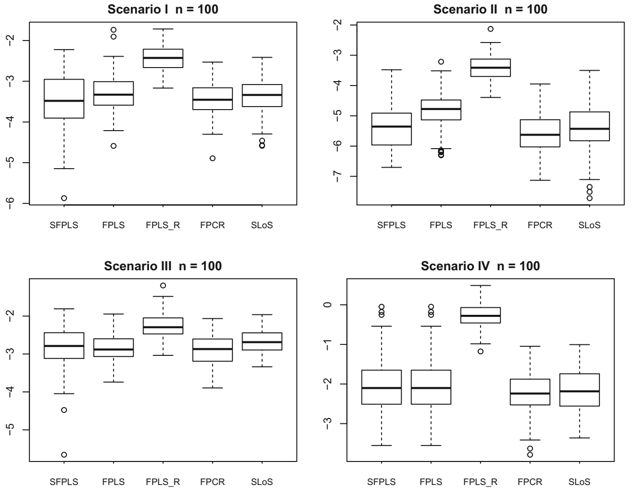

Fig. 6 Boxplots of the logarithmic integrated squared error, ISE_1, defined on the nonzero sub-regions for $\hat{\beta}(t)$ based on 100 simulation replications with $n = 100$

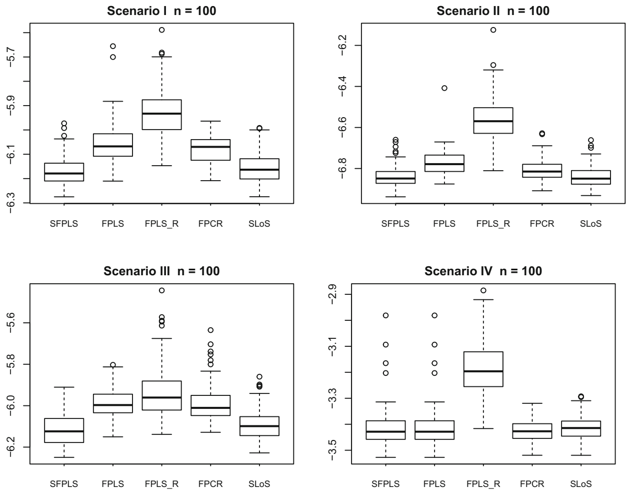

Fig. 7 Boxplots of the logarithmic prediction mean squared errors, PMSE, on the test data based on 100 simulation replications with $n = 100$

of ISE_1 of the estimators. It is observed that the proposed SFPLS method, the FPLS method, the FPCR method, and the SLoS method have similar performances. In Scenario IV when there is no zero sub-region, the SFPLS method has nearly identical results as the FPLS method, as expected. It is observed that the FPLS_R approach exhibits relatively larger errors. In terms of prediction, Fig. 7 suggests that, for scenarios I, II, and III, the proposed SFPLS method has the best performance. The SLoS method has a very similar performance to the proposed method. For Scenario IV, all the methods except FPLS_R are comparable. The results for $n = 500$ presented in the supplementary document are similar.

Among the dimension reduction methods SFPLS, FPLS, FPLS_R and FPCR, in general the SFPLS method selects the least number of components. For example, the average number of components selected by the SFPLS, FPLS, FPLS_R and FPCR methods for Scenario I and $n = 100$ are 3.42, 5.28, 6.86 and 6.06, respectively.

To visualize the performance of the proposed SFPLS method, Fig. 8 compares the estimated $\hat{\beta}(t)$ for various methods. We can see that in the nonzero sub-regions, the SFPLS method, the FPLS method, the FPCR method, and the SLoS method provide smooth estimates for the slope function. However, for scenarios I and II, only the SFPLS and SLoS estimates are capable of correctly identifying a major portion of the zero sub-regions. For scenario III, only the SFPLS method correctly identifies the two disconnected zero sub-regions. We can see that the SFPLS estimators might be discontinuous, which helps detect the zero sub-regions more accurately. For Scenario IV where there are no zero sub-regions, the SFPLS and FPLS estimates are identical. It also shows that the FPLS_R method might not have sufficient roughness regularization of the estimated slope function, which leads to less favorable performance. In Fig. 9, we present the SFPLS estimates for all simulation replicates with $n = 100$. The plots suggest that the SFPLS method locates the zero sub-regions with considerable accuracy and does a good job of estimating the slope functions.

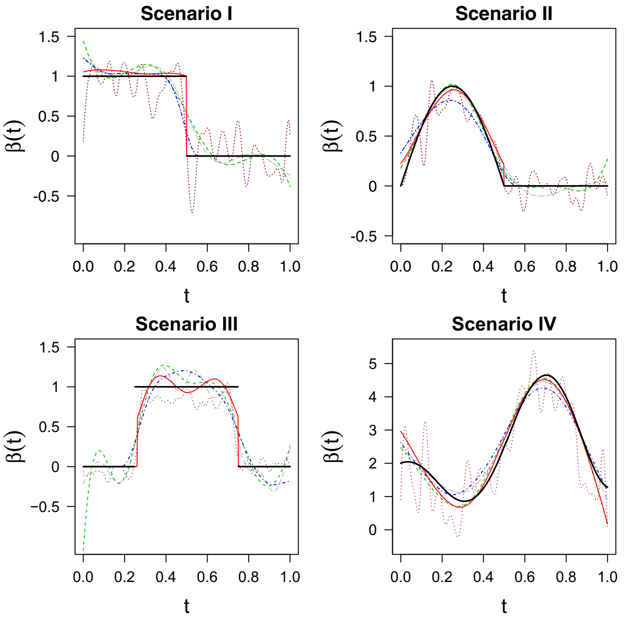

Fig. 8 Estimated $\hat{\beta}(t)$ for various methods (———, the proposed SFPLS method; - - - - -, the FPLS method; ··········, the FPLS_R method; — — —, the FPCR method; -·-·-·-, the SLoS method; ———, the true $\beta(t)$) in a randomly selected simulation replicate with $n = 100$

Next, we investigate the computational costs of all methods. Table 1 compares the computation time (in seconds) in the example with Scenario I for $\beta(t)$ and $n = 100$. All results

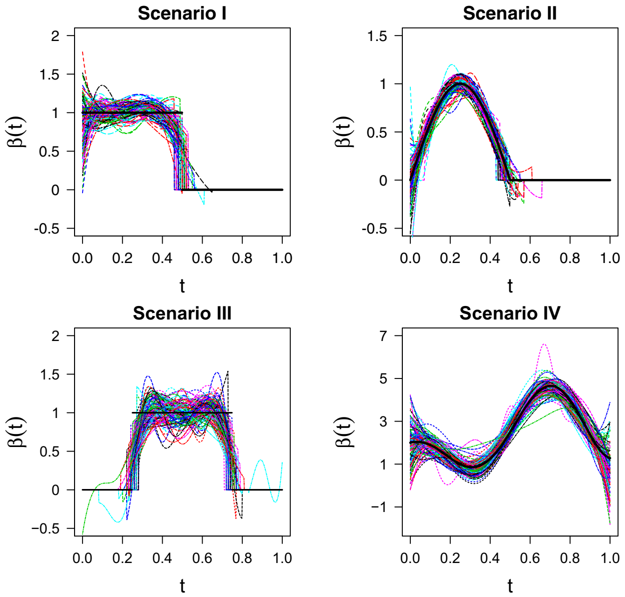

Fig. 9 Estimated slope functions $\hat{\beta}(t)$ by the SFPLS method (- - - - -) with $n = 100$ along with the true $\beta(t)$ (———)

Table 1 The computation time (in seconds) of all methods for Scenario I and $n = 100$

SFPLS	FPLS	FPLS$_R$	FPCR	SLoS
24.21	0.90	8.39	2.98	22.50

presented are the average of 20 independent trials. The experiments were conducted using a single processor Intel Core i5 @2.7 GHz (this processor has 2 cores and 4 threads) and 8GB of RAM, equipped with macOS Sierra version 10.12.6. The results of other scenarios for $\beta(t)$ are similar and hence omitted. All the computation time includes the model fitting and tuning. Table 1 shows that the FPLS and FPCR methods are fast. The SFPLS and SLoS methods are slower because they are more complicated methods that are capable of identifying the zero sub-regions. We can optimize the SFPLS and SLoS methods by parallel computing.

4 Applications

4.1 OSB furnish data

We apply our method to a novel dataset provided by FPInnovations which collected wood material from Canadian OSB mills. In total, 60 cookies from 60 different logs were collected with 10 from each of the following types of logs: aspen with core rot, sound aspen with bark, balsam poplar with core rot, balsam poplar with bark, birch with core rot, and birch with bark. Then the bark and rot were removed from all samples and segregated into separate sample bags. Sound wood, rot, and bark material from each species were ground into powder. Each sample was then oven-dried to minimize the measurement error due to moisture variation and stored in a sealed sample bag. In the end, 182 mixtures with different proportions of sound wood were prepared.

The powder samples were scanned by a benchtop Vis–NIR (visible and near-infrared) spectrometer ASD 5000 from Analytical Spectral Devices for acquiring spectra. Vis–NIR spectroscopy is an industrial proven measurement technique for classifying and quantifying the composition and properties of organic materials including wood. In the visible range from 390 to 700 nm the color composition of the measured sample was recorded. The records in the NIR range from 700 to 2500 nm reveal special interactions between light at specific wavelengths and target sample molecular structures, from which specific sample constituents can be chemically identified and quantified. Specifically, for each sample, a spectral file was generated comprised of log inverse reflectance versus wavelength for 2150 individual wavelengths ranging from 350 nm to 2500 nm. For each of the 182 samples, three spectra replicates were acquired.

The objective of this study is to examine the relation between the spectral trajectories and the proportions of the sound wood in OSB fines samples. In the functional linear regression model, for the ith fines mixture sample, the response Y_i is the proportion of sound wood content and $X_i(t), t \in [350, 2500]$ is its spectral curve. Both Y_i and $X_i(t)$ are centered such that $\mathbf{E}Y_i \equiv 0$ and $\mathbf{E}X_i(t) \equiv 0$. Figure 10a displays 10 randomly selected smoothed centered spectral curves.

Figure 10b depicts estimates for $\beta(t)$ obtained by the proposed SFPLS approach and the conventional FPLS method. The SFPLS method uses cubic B-spline basis functions with 101 equally spaced knots. We set $\kappa = 0.1$ and $\delta = 0.1$, and select γ, λ and the number of components K by BIC. The SFPLS estimate for the slope function is zero over wavelengths 587–2285nm, which suggests the relationship between the sound wood proportions and spectra in the low

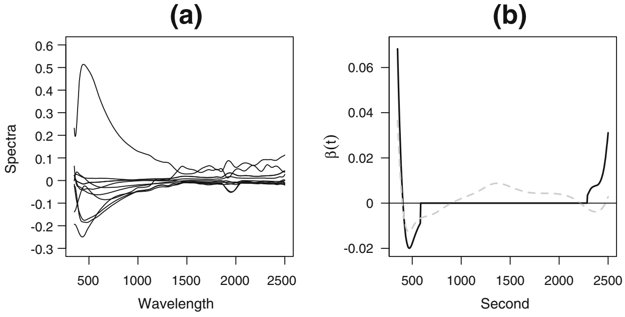

Fig. 10 **a** 10 randomly selected smoothed centered spectral curves. **b** Estimated $\hat{\beta}(t)$ using the conventional FPLS method (- - - - -) and the proposed SFPLS approach (———)

and high ends of the wavelengths. The FPLS method provides a similar estimate without giving clear sub-regions on which the spectra are not related to the sound wood proportions.

4.2 Particulate matter emissions data

We further illustrate our proposed procedure by analyzing the particulate matter emissions (PM) data, which was studied in Asencio et al. (2014), Hall and Hooker (2016), and Guan et al. (2020). The data are detailed in the Coordinating Research Councils E55/E59 research project (Clark et al. 2007). In the experiment, drivers drove the trucks through a pre-set series of driving cycles on a chassis dynamometer test bed which was employed to simulate inertia, wind drag, and tire rolling resistance. The particulate matter emissions were measured every second by an emission analyzer which was attached to the truck exhaust pipe. The engine acceleration of diesel trucks was also collected. Our interest here is to estimate the effects of the past engine acceleration on the current particulate matter emission. Intuitively, we expect earlier engine acceleration to have a smaller impact on the current particulate matter emission.

To remove dependences in the data, we follow Hall and Hooker (2016) to use PM observation every 10 s after the first 120 s. The response is the logarithm of the PM measured every 10 s after the first 120 s and the functional covariates are the engine acceleration for the past 60 s. We center both the PM and engine acceleration. Figure 11a illustrates 10 randomly selected smoothed centered engine acceleration curves against the backward time, in which 0 means the current time.

The proposed SFPLS approach uses cubic B-spline basis functions with 101 equally spaced knots with $\kappa = 0.1$ and $\delta = 0.1$. The smoothing parameter γ, the sparse parameter λ, and the number of components K are chosen by BIC. Figure 11b presents the SFPLS and FPLS estimates for $\beta(t)$. The FPLS estimate suggests that the engine acceleration greatly influences the PM for the past 20 s. Compared to the FPLS method, the SFPLS method provides a more insightful and interpretable result. The SFPLS estimate indicates a positive relationship that tapers to zero from second 0 to 16. This suggests that the past engine acceleration has contribution to predicting the current PM for no longer than 16 s.

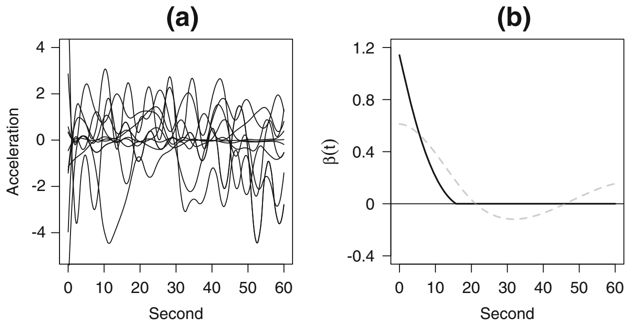

Fig. 11 **a** 10 randomly selected smoothed centered acceleration curves. **b** Estimated $\hat{\beta}(t)$ using the conventional FPLS method (- - - - -) and the proposed SFPLS approach (————)

5 Conclusion

Motivated by an analysis of a novel dataset provided by FPInnovations, we proposed a sparse functional partial least squares regression to simultaneously achieve locally sparse estimates for the functional least squares bases and a locally sparse estimate for the slope function in a functional linear regression model. The proposed method is effective in identifying nonactive sub-regions of the slope function. In the fitting procedure, the important tuning parameters include the smoothing parameter, the shrinkage parameter, and the number of components. We select these important tuning parameters via BIC.

Compared with the existing methods, the proposed method is able to provide a locally sparse estimate for the slope function via the functional partial least squares method. From our simulation studies, we found that the proposed SFPLS method improves the conventional FPLS method and performs well for continuous/discontinuous and locally sparse/nonsparse functions. In addition, the SFPLS estimators might be discontinuous, which helps detect the zero sub-regions more accurately. The real data applications show that the proposed method provides more interpretable estimates of the slope function.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-021-10066-y.

Acknowledgements The authors would like to thank the Editor, the Associate Editor, and two reviewers for their valuable comments, which are very helpful to improve this work. This work was supported by a discovery grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) to J. Cao.

References

Asencio, M., G. Hooker, and H. O. Gao (2014). Functional convolution models. Statistical Modelling 14(4), 315–335

Boulesteix, A.-L. and K. Strimmer (2006). Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. Briefings in Bioinformatics 8(1), 32–44

Cardot, H., F. Ferraty, and P. Sarda (2003). Spline estimators for the functional linear model. Statistica Sinica 13, 571–591

Chun, H., Keleş, S.: Sparse partial least squares regression for simultaneous dimension reduction and variable selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**(1), 3–25 (2010)

Clark, N., Gautam, M., Wayne, W., Lyons, D., Thompson, G., Zielinska, B.: Heavy-duty vehicle chassis dynamometer testing for emissions inventory, air quality modeling, source apportionment and air toxics emissions inventory: E55/59 all phases. Technical report, Coordinating Research Council, Alpharetta (2007)

Cook, R. D. and L. Forzani (2019). Partial least squares prediction in high-dimensional regression. The Annals of Statistics 47(2), 884–908

de Boor, C.: A practical Guide to Splines. Springer-Verlag, New York (2001)

Delaigle, A., Hall, P.: Achieving near perfect classification for functional data. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **74**(2), 267–286 (2012)

Delaigle, A. and P. Hall (2012b). Methodology and theory for partial least squares applied to functional data. The Annals of Statistics 40(1), 322–352

Escabias, M., A. M. Aguilera, and M. J. Valderrama (2007). Functional PLS logit regression model. Computational Statistics and Data Analysis 51(10), 4891–4902

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association 96(456), 1348–1360

Frank, L. E. and J. H. Friedman (1993). A statistical view of some chemometrics regression tools. Technometrics 35(2), 109–135

Garthwaite, P. H. (1994). An interpretation of partial least squares. Journal of the American Statistical Association 89(425), 122–127

Guan, T., Z. Lin, and J. Cao (2020). Estimating truncated functional linear models with a nested group bridge approach. Journal of Computational and Graphical Statistics 29(3), 620–628

Hall, P., Hooker, G.: Truncated linear models for functional data. Journal of Royal Statistical Society, Series B (Statistical Methodology) **78**(3), 637–653 (2016)

Helland, I. S. (1990). Partial least squares regression and statistical models. Scandinavian Journal of Statistics 17(2), 97–114

James, G. M., J. Wang, and J. Zhu (2009). Functional linear regression that's interpretable. The Annals of Statistics 37(5A), 2083–2108

Krämer, N. and M. Sugiyama (2011). The degrees of freedom of partial least squares regression. Journal of the American Statistical Association 106(494), 697–705

Lin, Z., J. Cao, L. Wang, and H. Wang (2017). Locally sparse estimator for functional linear regression models. Journal of Computational and Graphical Statistics 26(2), 306–318

Lin, Z., L. Wang, and J. Cao (2016). Interpretable functional principal component analysis. Biometrics 72, 846–854

Martens, H., Næs, T.: Multivariate Calibration. John Wiley & Sons, New York (1992)

Marx, B.D., Eilers, P.H.C.: Generalized linear regression on sampled signals and curves: A P-spline approach. Technometrics **41**(1), 1–13 (1999)

Nie, Y., Cao, J.: Sparse functional principal component analysis in a new regression framework. Computational Statistics and Data Analysis **152**, 107016 (2020)

Nie, Y., L. Wang, B. Liu, and J. Cao (2018). Supervised functional principal component analysis. Statistics and Computing 28(3), 713–723

Preda, C. and G. Saporta (2005). PLS regression on a stochastic process. Computational Statistics and Data Analysis 48, 149–158

Preda, C., Saporta, G., Lévéder, C.: PLS classification of functional data. Comput. Statistics **22**(2), 223–235 (2007)

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

Sang, P., L. Wang, and J. Cao (2017). Parametric functional principal component analysis. Biometrics 73, 802–810

Schwartz, W. R., Kembhavi, A., Harwood, D., Davis, L. S.: Human detection using partial least squares analysis. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 24–31 (2009)

Shi, H., J. Dong, L. Wang, and J. Cao (2021). Functional principal component analysis for longitudinal data with informative dropout. Statistics in Medicine 40, 712–724

Wold, H.: Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach. J. Appl. Probab. **12**(S1), 117–142 (1975)

Zhou, J., N.-Y. Wang, and N. Wang (2013). Functional linear model with zero-value coefficient function at sub-regions. Statistica Sinica 23, 25–50