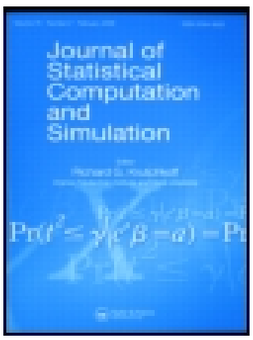

Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

Smooth and locally sparse estimation for multiple-output functional linear regression

Kuangnan Fang, Xiaochen Zhang, Shuangge Ma & Qingzhao Zhang

To cite this article: Kuangnan Fang, Xiaochen Zhang, Shuangge Ma & Qingzhao Zhang (2019): Smooth and locally sparse estimation for multiple-output functional linear regression, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2019.1680676

To link to this article: https://doi.org/10.1080/00949655.2019.1680676

Published online: 22 Oct 2019.

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
https://doi.org/10.1080/00949655.2019.1680676

Smooth and locally sparse estimation for multiple-output functional linear regression

Kuangnan Fang[a,b], Xiaochen Zhang[a], Shuangge Ma[c] and Qingzhao Zhang[a,b,d]

[a]Department of Statistics, School of Economics, Xiamen University, Xiamen, People's Republic of China; [b]Key Laboratory of Econometrics, Ministry of Education, Xiamen University, Xiamen, People's Republic of China; [c]Department of Biostatistics, Yale School of Public Health, New Haven, CT, USA; [d]The Wang Yanan Institute for Studies in Economics, Xiamen University, Xiamen, People's Republic of China

ABSTRACT

Functional data analysis has attracted substantial research interest and the goal of functional sparsity is to produce a sparse estimate which assigns zero values over regions where the true underlying function is zero, i.e. no relationship between the response variable and the predictor variable. In this paper, we consider a functional linear regression model that explicitly incorporates the interconnections among the responses. We propose a locally sparse (i.e. zero on some subregions) estimator, multiple-smooth and locally sparse (mSLoS) estimator, for coefficient functions base on the interconnections among the responses. Simulations show excellent numerical performance of the proposed method in terms of the estimation of coefficient functions especially the coefficient functions are same for all multivariate responses. Practical merit of this modelling is demonstrated by one real application and the prediction shows significant improvements.

ARTICLE HISTORY
Received 2 February 2019
Accepted 11 October 2019

KEYWORDS
Functional data analysis; locally sparse; functional linear multivariate regression

1. Introduction

Functional data analysis has attracted substantial research interest. In functional data analysis, functional linear regression (FLR) is a popular technique when predictions themselves are functions. Historically, FLR originates from the ordinary linear regression with a large number of predictors. It has consequently been thoroughly studied and extensively applied. A non-exhaustive list of recent works includes the followings [1–7].

In this article, we consider the functional linear regression for multivariate responses. Let $\mathbf{Y} = \{Y_1, \ldots, Y_p\}$ be the response vector and $X(t)$ be the functional predictor observed at a dense grid of points. Consider the following functional linear model:

$$Y_j = \mu_j + \int_0^T X(t)\beta_j(t)\,\mathrm{d}t + \epsilon_j, \quad j = 1, \ldots, q, \tag{1}$$

where μ_j is the intercept term, $\beta_j(t)$ is the unknown smooth coefficient function, and ϵ_j is the random error for the jth response. If on a subregion $I \subset [0, T]$, $\beta_j(t) = 0$ for every

CONTACT Qingzhao Zhang qzzhang@xmu.edu.cn

$t \in I$, then $X(t)$ has no contribution to Y_j on the interval I. In light of this observation, an estimate of $\beta_j(t)$ improves the interpretability of the model and is practically appealing, if it not only yields weights of the contribution of $X(t)$ over the entire domain but also locates subregions where $X(t)$ has no statistically significant contribution to Y_j.

This estimate of β_j mentioned above is called the locally sparse estimate [8,9]. Although the literature on FLR is abundant, little has been done on interpretability and locally sparse modelling, especially on the multivariate response. James et al. [10] proposed 'FliRTI' to achieves local sparseness by employing L_1 penalty on the coefficient function and its first several derivatives at some discrete grid points. Zhou et al. [11] pointed out one drawback of FliRTI method that the produced estimate possesses large variation. When the grid size is small, the numerical solution is unstable, while when the grid size is large, FLiRTI method tends to overparameterize the model. To overcome this background, Zhou et al. [11] proposed an alternative locally sparse estimator obtained in two stages. Lin et al. [12] proposed a simple one-stage procedure that yields a smooth and locally sparse estimator of the coefficient function and they call it 'smooth and locally sparse (SLoS) estimator '. All methods mentioned above are done with a univariate response, while we consider the multiple-output FLR in this paper.

In multivariate regression, the idea of using information from different responses to improve estimation is not new. Previous work has been done on scalar multivariate regression. Breiman and Friedman [13] proposed Curds and Whey method, which used optimal linear combination of least squares predictions as the predictors. Rothman et al. [14] proposed multivariate regression with covariance estimation, leveraged correlation in unexplained variation to improve estimation. Peng et al. [15] proposed regularized multivariate regression for identifying master predictors, which was motivated by investigating the regulatory relationships among different biological molecules based on multiple types of high-dimensional genomic data. Rai et al. [16] proposed a multiple-output regression model that allows leveraging both output structure and task structure with output structure and task structure learned from the data. It relies on a priori information about valuable predictors. They imposed a group L_1 and L_2 norm, across responses, on all covariates not prespecified as being useful predictors. Bradley and Ben [17] proposed a method for simultaneously estimating regression coefficients and clustering response variables in a multivariate regression model, to increase prediction accuracy and give insights into the relationship between response variables. Shi et al. [18] proposed a novel method, Variational Inference for Multiple Correlated Outcomes, for joint analysis of multiple traits in Genome-Wide Association Studies and a variational Bayesian expectation–maximization algorithm was used to ensure computational efficiency. In this paper, we assume that if Y_k and Y_j are tightly connected, then their regression coefficient profiles $\beta_k(t)$ and $\beta_j(t)$ should be similar on local sparsity. Although the literature on scalar multivariate regression is abundant, little has been done on functional multivariate regression.

Based on the above discussions, we aim to develop locally sparse estimator for the coefficient function $\beta_j(t)$ for $j = 1, \ldots, q$, while effectively accommodate correlations among the multivariate responses. In this article, we consider a combination of the SLoS and Laplacian quadratic as the penalty function. We call the proposed method 'multiple-SLoS' (m-SLoS). This method uses the SLoS for encouraging locally sparse and Laplacian

quadratic penalty for promoting similar local sparsity among coefficient functions associated with the interconnections among the responses. Note that the Laplacian quadratic penalty here is imposed on functions, and is different from that in Huang et al. [19], which promotes similarities among regression coefficients.

The later sections are organized as follows. The model setting and methodology are described in Section 2 as well as the computational algorithm. In Section 3, we present simulation studies under the four different scenarios to assess finite performance of our proposed method. In Section 4, we apply the proposed method to Tecator data. The article concludes with a discussion in Section 5.

2. Methods

2.1. The model setting and methodology

Under the smoothness condition, we approximate $\beta_j(\cdot)$ using the B-spline basis expansion. Given M_n evenly-spaced knots, $0 = t_0 < t_1 < t_2 < \cdots < t_{M_n-1} < t_{M_n} = T$, let $I_k = [t_{k-1}, t_k]$ for $k = 1, \ldots, M_n$. Associated with this set of knots, there are $(M_n + d)$ B-spline basis functions, $\boldsymbol{B}(t) = (B_1(t), \ldots, B_{M_n+1}(t))^\top$, each of which is a piecewise polynomial of degree d with support at most $d+1$ subintervals I_k. Then

$$\beta_j(t) = \sum_{l=1}^{M_n+d} b_{j,l} B_l(t) + r_j(t),$$

where $r_j(t)$ is an approximation error that is uniformly bounded on $[0, T]$ with the bound going to 0 as M_n goes to infinity. Let U be an $n \times (M_n + d)$ matrix with entries $u_{i,j} = \int_0^T X_i(t) B_j(t)\, \mathrm{d}t$ and $U = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)^\top$. Moreover, set $\boldsymbol{b} = (\boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_p^T)^\top$ and $\boldsymbol{b}_j = (b_{j,1}, \ldots, b_{j,M_n+d})^\top$. Then model (1) can be written as

$$\boldsymbol{y}_j = \mu_j + U\boldsymbol{b}_j + \varepsilon_j, \quad j = 1, \ldots, q,$$

where $\boldsymbol{y}_j = (y_{1j}, \ldots, y_{nj})^\top$, and the error $\varepsilon_j = (\varepsilon_{1j}, \ldots, \varepsilon_{nj})^\top$ satisfies $\varepsilon_{ij} = \epsilon_{ij} + \int_0^T X_i(t) r_j(t)\, \mathrm{d}t$. In this section, we adopt the least squares objective functions

$$L(\boldsymbol{\mu}, \boldsymbol{b}) = \frac{1}{nq} \sum_{j=1}^{q} \left\| \boldsymbol{y}_j - \mu_j \mathbf{1}_{n\times 1} - U\boldsymbol{b}_j \right\|^2,$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_q)^\top$, and $\|\cdot\|$ is the l_2 norm.

Moreover, suppose the adjacency matrix is $A = (a_{ij}, 1 \leqslant i, j \leqslant q)$ for the responses and we want to accommodate the correlation structure to promote similar local sparsity among coefficient functions associated with the interconnections among the responses. The idea of using Laplacian quadratic penalty to promote similarities among coefficients was not new. Huang et al. [19] proposed Laplacian quadratic penalty for variable selection and estimation that explicitly incorporates the correlation patterns among predictors. Shi et al. [20] proposed a sparse double Laplacian shrinkage method which jointly models the effects of multiple CNAs on multiple GEs. Wu et al. [21] borrowed the idea of Laplacian penalty and proposed a method to comprehensively accommodate multiple challenging characteristics

of GE–CNV modelling. In this paper, we extend the usage of Laplacian quadratic penalty to multiple-output FLR. Our goal is to promote similar locally sparse among coefficient functions associated with the interconnections among the responses. To accommodate the correlation structure, we propose $\lambda \sum_{1\leqslant k<j\leqslant q} |a_{kj}| \int_0^T [\beta_k(t) - \operatorname{sgn}(a_{kj})\beta_j(t)]^2 \,\mathrm{d}t$.

Note that

$$\int_0^T \left[\beta_k(t) - \operatorname{sgn}(a_{kj})\beta_j(t)\right]^2 \,\mathrm{d}t \approx (\boldsymbol{b}_k - \operatorname{sgn}(a_{kj})\boldsymbol{b}_j)^\top \Theta (\boldsymbol{b}_k - \operatorname{sgn}(a_{kj})\boldsymbol{b}_j),$$

where $\Theta = \int_0^T \boldsymbol{B}(t)\boldsymbol{B}(t)^\top \,\mathrm{d}t$. Let $D = \operatorname{diag}(d_1, \ldots, d_q)$, where $d_k = \sum_{j=1}^q |a_{kj}|$. Define $L = D - A$, which can be easily proved to be a positive semi-definite matrix. Then

$$\boldsymbol{b}^\top (L \otimes \Theta)\boldsymbol{b} = \sum_{1\leqslant j<k\leqslant p} \left|a_{jk}\right| (\boldsymbol{b}_k - \operatorname{sgn}(a_{kj})\boldsymbol{b}_j)^\top \Theta (\boldsymbol{b}_k - \operatorname{sgn}(a_{kj})\boldsymbol{b}_j),$$

where $\otimes$ is the Kronecker product. More related discussions refer to Huang et al. [19].

Lin et al. [12] developed a SLoS estimator for coefficient function based on 'fSCAD' penalty: $(1/T)\int_0^T p_\lambda(|\beta_j(t)|)\,\mathrm{d}t$. Here $p_\lambda(t)$ is the SCAD penalty, where λ is a data-dependent tuning parameter. From Theorem 1 in their paper,

$$\begin{aligned}\frac{1}{T}\int_0^T p_\lambda(\left|\beta_j(t)\right|)\,\mathrm{d}t &= \lim_{M_n\to\infty} \frac{1}{M_n} \sum_{k=1}^{M_n} p_\lambda\left(\sqrt{\frac{M_n}{T}\int_{t_{k-1}}^{t_k} \beta_j^2(t)\,\mathrm{d}t}\right) \\ &= \lim_{M_n\to\infty} \frac{1}{M_n} \sum_{k=1}^{M_n} p_\lambda\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}\right),\end{aligned}$$

where W_k is an $(M_n + d) \times (M_n + d)$ matrix with entries $w_{uv} = \int_{t_{k-1}}^{t_k} B_u(t)B_v(t)\,\mathrm{d}t$ if $k \leqslant u, v \leqslant k + d$ and zeros otherwise. When M_n is relative large, the estimator usually exhibits excessive variability. A popular approach to rectify the variability is to add a roughness penalty on $\beta_j(t) = \boldsymbol{B}(t)^\top \boldsymbol{b}_j$. For example,

$$\eta_n \int_0^T \beta''(t)^2 \,\mathrm{d}t = \eta_n \boldsymbol{b}_j^\top V \boldsymbol{b}_j,$$

where V is an $(M_n + d) \times (M_n + d)$ matrix with entries

$$v_{ij} = \int_0^T \frac{\mathrm{d}^2 B_i(t)}{\mathrm{d}t^2} \frac{\mathrm{d}^2 B_j(t)}{\mathrm{d}t^2}\,\mathrm{d}t.$$

Based on the above discussion, we propose the following objective function for functional linear multivariate regression:

$$Q(\boldsymbol{\mu}, \boldsymbol{b}) = L(\boldsymbol{\mu}, \boldsymbol{b}) + \sum_{j=1}^q \sum_{k=1}^{M_n} p_{\lambda_1}\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}\right) + \lambda_2 \sum_{j=1}^q \boldsymbol{b}_j^\top V \boldsymbol{b}_j + \lambda_3 \boldsymbol{b}^\top (L \otimes \Theta)\boldsymbol{b}. \tag{2}$$

In (2), the first part is the usual least squares objective function. The second part is used to encourage the local sparsity of coefficient functions. The third part is a roughness penalty, which is a popular approach to rectify the variability of coefficient functions. The last part is Laplacian quadratic penalty, which can promote similar local sparsity among coefficient functions associated with the interconnections among the responses. We called this methods as 'multiple-SLoS' (m-SLoS). The estimator by minimizing (2) enjoys smooth and locally sparse properties. In the next section, we will show the algorithm to solve this problem.

2.2. Computational algorithm

In this section, we will discuss the algorithm to minimizing (2). Before solving this optimization problem, we have to get the adjacency matrix A. If we have prior information of A, we can just use it. If we do not have prior information, we could use the data to calculate the adjacency matrix among responses. More related discussions refer to Huang et al. [19]. In this paper, we use the correlation matrix of responses as A.

When $u \approx u^{(0)}$,the LQA of SCAD function $p_\lambda(\mu)$ is

$$p_\lambda(|\mu|) \approx p_\lambda\left(\left|\mu^{(0)}\right|\right) + \frac{1}{2}\frac{p_\lambda{}'\left(\left|\mu^{(0)}\right|\right)}{\left|\mu^{(0)}\right|}(\mu^2 - \mu^{(0)2}).$$

Then given some initial estimate $\boldsymbol{b}_j^{(0)}$, for $\boldsymbol{b}_j^{(0)} \approx \boldsymbol{b}_j$, we have

$$\sum_{k=1}^{M_n} p_{\lambda_1}\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}\right) \approx \frac{1}{2}\sum_{k=1}^{M_n} \frac{p_{\lambda_1}{}'\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}\right)}{\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}} \frac{\boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}{T/M_n} + G(\boldsymbol{b}_j^{(0)}),$$

where

$$G(\boldsymbol{b}_j^{(0)}) = \sum_{k=1}^{M_n} p_{\lambda_1}\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}\right) - \frac{1}{2}\sum_{k=1}^{M_n} p_{\lambda_1}{}'\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}\right)\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}$$

only depends on the initial estimate $\boldsymbol{b}_j^{(0)}$.

Let

$$W^{(0)} = \frac{1}{2}\sum_{k=1}^{M_n} \frac{p_{\lambda_1}{}'\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}\right)}{\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^{(0)\top} W_k \boldsymbol{b}_j^{(0)}}} W_k,$$

then we have

$$\sum_{k=1}^{M_n} p_{\lambda_1}\left(\sqrt{\frac{M_n}{T}\boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}\right) \approx \boldsymbol{b}_j^\top W^{(0)} \boldsymbol{b}_j + G(\boldsymbol{b}_j^{(0)}).$$

Recall the objective function is

$$Q(\boldsymbol{\mu}, \boldsymbol{b}) = L(\boldsymbol{\mu}, \boldsymbol{b}) + \sum_{j=1}^{q} \sum_{k=1}^{M_n} p_{\lambda_1}\left(\sqrt{\frac{M_n}{T} \boldsymbol{b}_j^\top W_k \boldsymbol{b}_j}\right) + \lambda_2 \sum_{j=1}^{q} \boldsymbol{b}_j^\top V \boldsymbol{b}_j + \lambda_3 \boldsymbol{b}^\top (L \otimes \Theta) \boldsymbol{b},$$

where $L(\boldsymbol{\mu}, \boldsymbol{b}) = (1/nq) \sum_{j=1}^{q} \|\boldsymbol{y}_j - \mu_j \mathbf{1}_{n\times 1} - U\boldsymbol{b}_j\|^2$, W_k is an $(M_n + d) \times (M_n + d)$ matrix with entries $w_{uv} = \int_{t_{k-1}}^{t_k} B_u(t) B_v(t)\, \mathrm{d}t$ if $k \leqslant u, v \leqslant k + d$ and zeros otherwise, V is an $(M_n + d) \times (M_n + d)$ matrix with entries

$$v_{ij} = \int_0^T \frac{\mathrm{d}^2 B_i(t)}{\mathrm{d}t^2} \frac{\mathrm{d}^2 B_j(t)}{\mathrm{d}t^2}\, \mathrm{d}t.$$

Now we get

$$Q(\boldsymbol{\mu}, \boldsymbol{b}) = L(\boldsymbol{\mu}, \boldsymbol{b}) + \sum_{j=1}^{q} (\boldsymbol{b}_j^\top W^{(0)} \boldsymbol{b}_j + G(\boldsymbol{b}_j^{(0)})) + \lambda_2 \sum_{j=1}^{q} \boldsymbol{b}_j^\top V \boldsymbol{b}_j + \lambda_3 \boldsymbol{b}^\top (L \otimes \Theta) \boldsymbol{b}.$$

Let $R(\boldsymbol{b}_j)$ denote the terms that contain $\boldsymbol{b}_j$, we have

$$\begin{aligned} R(\boldsymbol{b}_j) = {} & \frac{1}{nq} \left\| \boldsymbol{y}_j - \mu_j 1_{n\times 1} - U b_j \right\|^2 + \boldsymbol{b}_j^\top W^{(0)} \boldsymbol{b}_j + \lambda_2 \boldsymbol{b}_j^\top V \boldsymbol{b}_j \\ & + 2\lambda_3 \sum_{l \neq j} \boldsymbol{b}_l^\top L_{jl} \Theta \boldsymbol{b}_l + \boldsymbol{b}_j^\top L_{jj} \Theta \boldsymbol{b}_j. \end{aligned}$$

Differentiating $R(\boldsymbol{b}_j)$ with respect to $\boldsymbol{b}_j$ and setting it to zero, we have the following equation:

$$\left[U^\top U + nqW^{(0)} + nq\lambda_2 V + nq\lambda_3 L_{jj} \Theta^\top \right] \boldsymbol{b}_j = U^\top \boldsymbol{y}_j - nq\lambda_3 \sum_{l \neq j} L_{jl} \Theta^\top \boldsymbol{b}_l$$

with the solution

$$\hat{\boldsymbol{b}}_j = \left[U^\top U + nqW^{(0)} + nq\lambda_2 V + nq\lambda_3 L_{jj} \Theta^\top \right]^{-1} \left[U_j^\top \boldsymbol{y}_j - nq\lambda_3 \sum_{l \neq j} L_{jl} \Theta^\top \boldsymbol{b}_l \right].$$

In summary, we have the following algorithm to compute $\hat{\boldsymbol{b}}_j$ and obtain the estimator $\hat{\beta}_j(t) = \boldsymbol{B}(t)^\top \hat{\boldsymbol{b}}_j$ for $j = 1, 2, \ldots, q$.

Step 1: for $j = 1, 2, \ldots, q$,

(a) Compute the initial estimate $\hat{\boldsymbol{b}}_j^{(0)} = [U^\top U + n\lambda_2 V]^{-1} U^\top \boldsymbol{y}_j$.
(b) Given $\hat{\boldsymbol{b}}_j^{(i)}$, compute $W^{(i)}$ and $\hat{\boldsymbol{b}}_j^{(i+1)} = [U^\top U + nW^{(0)} + n\lambda_2 V]^{-1} U^\top \boldsymbol{y}_j$.
(c) Repeat (b) until the convergence of $\hat{\boldsymbol{b}}_j$ is reached.

Step 2: Let the initial value $\hat{\boldsymbol{b}}_j^{(0)}$ be the value we get from *Step 1*.

Step 3: Given $\hat{\boldsymbol{b}}_j^{(i)}$ for $j = 1, 2, \ldots, q$, update

$$\hat{\boldsymbol{b}}_j^{(i+1)} = \left[U^\top U + nqW^{(0)} + nq\lambda_2 V + nq\lambda_3 L_{jj}\Theta^\top\right]^{-1}\left[U^T\boldsymbol{y}_j - nq\lambda_3\sum_{l\neq j} L_{jl}\Theta^\top\hat{\boldsymbol{b}}_j^{(i)}\right].$$

Step 4: Repeat *Step 3* until the convergence of $\hat{\boldsymbol{b}}_j$ is reached for $j = 1, 2, \ldots, q$.

Remark 2.1: We first calculate the SLoS estimator respectively in *Step 1*. Then we use this estimator as the initial value of our proposed method. The m-SLoS estimator is calculated in *Step 3*.

3. Simulation studies

In this section, we conduct simulation studies to evaluate the performance of the proposed method. We conduct a simulation study on the following function linear model:

$$Y_j = \mu_j + \int_0^1 X(t)\beta_j(t)\,\mathrm{d}t + \epsilon_j, \quad j = 1, \ldots, q,$$

where the true parameter $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_q)^\top = (1, \ldots, 1)^\top$. The covariate function $X_i(t) = \sum_j c_{ij}B_j(t)$ where the coefficients c_{ij} are generated from the standard uniform distribution on $[-5, 5]$ and each $B_j(t)$ is a B-spline basis function defined by order 5 and 50 equally spaced knots. We independently generate n datasets as the training set. The tune parameters, λ_1, λ_2 and λ_3, are selected using fivefold cross-validation. We have developed R code and made it publicly available at https://github.com/ruiqwy/m-slos.

To show the appearance of new method when β_js are same or similar, we present simulation studies under four different scenarios to assess the performance of our proposed model to SLoS. Coefficient functions for those four settings are plotted in Figure 1. In the first example, β_js are same. In the second example, β_js share the same zero intervals but the values of them are not same on nonzero intervals. As for the last two examples, β_js' zero intervals are similar but not same. Before we show the details of those four scenarios, we introduce some indexes to measure the performances of those estimations of coefficient functions.

The quality of estimates is measured by the integrated squared error (ISE) and integrated absolute error (IAE). ISE_0 and ISE_1 measure integrated squared errors between an estimated coefficient function $\hat{\beta}(t)$ and the true function $\beta(t)$ on null subregions and non-null subregions, respectively. IAE_0 and IAE_1 measure integrated absolute errors between an estimated coefficient function $\hat{\beta}(t)$ and the true function $\beta(t)$ on null subregions and non-null subregions, respectively. These indexes are described in more details below.

$$\mathrm{ISE}_0 = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{l_{0j}}\int_{N(\beta_j)}\left[\beta_j(t) - \hat{\beta}_j(t)\right]^2\mathrm{d}t, \quad \mathrm{ISE}_1 = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{l_{1j}}\int_{S(\beta_j)}\left[\beta_j(t) - \hat{\beta}_j(t)\right]^2\mathrm{d}t,$$

$$\mathrm{IAE}_0 = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{l_{0j}}\int_{N(\beta_j)}\left|\beta_j(t) - \hat{\beta}_j(t)\right|\mathrm{d}t, \quad \mathrm{IAE}_1 = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{l_{1j}}\int_{S(\beta_j)}\left|\beta_j(t) - \hat{\beta}_j(t)\right|\mathrm{d}t,$$

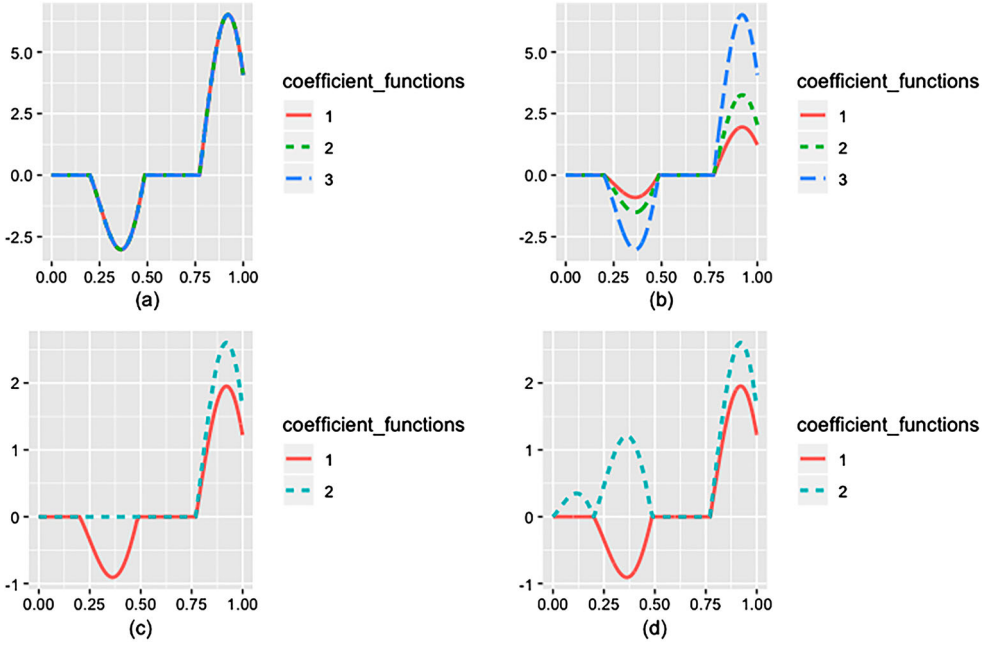

Figure 1. Coefficient functions in four settings: (a) Example 3.1; (b) Example 3.2; (c) Example 3.3; (d) Example 3.4.

where $\beta_j(t)$ is the true coefficient function, $\hat{\beta}_j(t)$ is the estimated coefficient function using SLoS or m-SLoS, $N(\beta_j)$ is the null subregions of $\beta_j(t)$, $S(\beta_j)$ is the non-null subregions of $\beta_j(t)$, l_{0j} is the length of null subregions $N(\beta_j)$ and l_{1j} is the length of non-null subregions $S(\beta_j)$.

In addition, to assess the performance of local sparsity detection, we used three numerical measures: the average percentage of intervals correctly identified (CI), the average percentage of true zero intervals correctly identified (CZ) and the average percentage of nonzero intervals correctly identified as zeros (CN). The bigger CI, CZ and CN are, the better the estimator is. For the best estimator, those three indexes are close to 1. We show the details of the four scenarios below.

Example 3.1: In this example, we set trivariate functional data with the same coefficient function. Here we set

$$\beta_1(t) = \beta_2(t) = \beta_3(t) = \begin{cases} \max\{0, -10\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 10\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1. \end{cases}$$

Then $\beta_j(t) = 0$ on $[0, 0.2] \cup [0.486, 0.771]$ for $j = 1, 2, 3$. The distribution of ϵ_j is set to be $N(0, \sigma^2)$. We set different errors to see the differences under different levels of error. The number of observations in the training set is n. The results of 100 Monte Carlo repetitions are showed in Table 1. From this table, we can find that the performances of estimations to those two methods both get better with the sample size n increase. The performances are better when the variance of ϵ_j is smaller.

Table 1. Results of Example 3.1 obtained from 100 Monte Carlo repetitions (with standard errors in parentheses).

σ^2	n	Method	CI (%)	CZ (%)	CN (%)	ISE_0 (1e−3)	ISE_1 (1e−3)	IAE_0 (1e−3)	IAE_1 (1e−3)
0.05	300	m-SLoS	89.45	78.35	99.91	1.41	9.22	8.56	57.76
			(5.11)	(10.66)	(0.26)	(0.90)	(4.34)	(5.47)	(16.39)
		slos	68.01	34.03	100.00	4.87	9.88	54.07	94.66
			(4.82)	(9.95)	(0.00)	(0.98)	(1.40)	(10.91)	(7.10)
	500	m-SLoS	90.99	81.50	99.92	1.35	9.58	7.25	57.19
			(3.24)	(6.79)	(0.24)	(0.73)	(7.90)	(3.39)	(20.52)
		slos	72.38	43.05	100.00	3.94	8.86	43.60	86.22
			(4.39)	(9.05)	(0.00)	(0.61)	(1.05)	(6.89)	(6.18)
	1000	m-SLoS	91.22	81.93	99.96	1.22	7.26	6.50	48.91
			(2.84)	(5.92)	(0.17)	(0.68)	(2.39)	(3.06)	(10.55)
		slos	76.85	52.26	100.00	3.32	7.91	35.94	77.48
			(4.19)	(8.65)	(0.00)	(0.52)	(0.72)	(5.29)	(4.63)
0.1	300	m-SLoS	71.65	44.75	96.97	17.72	62.26	44.85	157.10
			(2.71)	(7.14)	(2.05)	(10.12)	(35.99)	(15.32)	(58.02)
		slos	64.91	27.70	99.96	6.39	19.35	48.46	139.70
			(4.47)	(9.25)	(0.11)	(1.404)	(4.30)	(7.55)	(14.90)
	500	m-SLoS	82.09	63.13	99.96	2.46	11.62	16.70	68.45
			(5.43)	(11.31)	(0.22)	(1.30)	(11.08)	(8.96)	(24.50)
		slos	64.59	26.99	100.00	7.18	13.41	72.46	116.22
			(4.58)	(9.45)	(0.00)	(1.64)	(2.33)	(13.57)	(11.05)
	1000	m-SLoS	85.38	69.93	99.94	2.00	10.95	12.36	63.92
			(4.65)	(9.70)	(0.250)	(0.95)	(7.90)	(6.69)	(22.27)
		slos	67.38	32.74	100.00	5.95	12.39	59.09	104.92
			(4.52)	(9.33)	(0.00)	(1.01)	(1.64)	(9.47)	(8.03)

Example 3.2: In this example, we set trivariate functional data with coefficient functions share the same zero intervals. We want to see the performance of our new proposed method when the coefficient functions are not same but share the same zero intervals. The difference between this example and Example 3.1 is that the values of coefficient function on nonzero intervals are different. Here we set

$$\beta_1(t) = \begin{cases} \max\{0, -3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1, \end{cases}$$

$$\beta_2(t) = \begin{cases} \max\{0, -5\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 5\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1, \end{cases}$$

$$\beta_3(t) = \begin{cases} \max\{0, -10\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 10\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1. \end{cases}$$

Then $\beta_j(t) = 0$ on $[0, 0.2] \cup [0.486, 0.771]$ for $j = 1, 2, 3$. β_js share the same zero intervals but they are not same on nonzero intervals. The distribution of $\epsilon_j \overset{iid}{\sim} N(0, 0.05^2)$. The number of observations in training set is n. The results of 100 Monte Carlo repetitions are showed in Table 2. From this table, we can find that the performances of estimations to those two methods all get better with the sample size n increase. This results show that the newly proposed method also works if the coefficient functions share the same zero intervals though they are not same on the whole intervals.

Example 3.3: In this example, we set bivariate functional data with coefficient functions share similar zero intervals. One of the coefficient functions is more sparse than another

Table 2. Results of Examples 3.2 and 3.3 obtained from 100 Monte Carlo repetitions (with standard errors in parentheses).

n	Method	CI (%)	CZ (%)	CN (%)	ISE_0 (1e−3)	ISE_1 (1e−3)	IAE_0 (1e−3)	IAE_1 (1e−3)
				Example2				
300	m-SLoS	76.55	51.91	99.75	4.67	8.60	45.19	64.52
		(6.31)	(13.30)	(0.78)	(2.52)	(7.72)	(19.95)	(22.26)
	slos	61.84	21.35	99.96	8.04	12.35	81.32	117.52
		(4.31)	(8.91)	(0.10)	(2.60)	(2.54)	(19.45)	(12.63)
500	m-SLoS	87.21	73.65	99.99	2.11	8.55	10.83	54.72
		(4.05)	(8.36)	(0.14)	(0.75)	(3.83)	(4.60)	(14.31)
	slos	72.55	43.40	100.00	3.91	8.84	43.33	86.28
		(8.50)	(17.52)	(0.00)	(1.03)	(1.76)	(12.31)	(10.15)
1000	m-SLoS	88.57	76.43	100.00	2.01	7.82	9.95	49.65
		(2.94)	(6.07)	(0.00)	(0.51)	(1.46)	(3.74)	(7.31)
	slos	76.86	52.29	100.00	3.41	8.06	36.73	77.95
		(7.29)	(15.03)	(0.00)	(0.84)	(1.19)	(9.45)	(7.33)
				Example3				
300	m-SLoS	90.66	84.67	99.97	0.12	1.92	2.06	25.92
		(5.86)	(10.09)	(0.23)	(0.16)	(1.00)	(2.71)	(8.78)
	slos	66.43	77.11	99.88	0.48	3.17	9.05	50.45
		(5.26)	(12.93)	(0.18)	(0.47)	(1.31)	(6.95)	(9.55)
500	m-SLoS	93.29	88.69	99.97	0.12	1.57	2.05	21.36
		(3.65)	(6.30)	(0.24)	(0.14)	(0.91)	(2.67)	(7.39)
	slos	66.51	78.11	99.93	0.32	2.22	7.77	41.07
		(4.65)	(10.86)	(0.15)	(0.28)	(0.65)	(5.25)	(6.20)
1000	m-SLoS	96.03	93.14	99.98	0.05	1.11	0.69	16.25
		(1.73)	(2.76)	(0.17)	(0.06)	(0.76)	(0.85)	(6.62)
	slos	71.42	90.12	99.96	0.09	1.59	2.75	33.22
		(1.07)	(2.93)	(0.12)	(0.08)	(0.43)	(1.17)	(4.26)

one. We want to see whether m-SLoS will cause a misestimate of another coefficient function or not. Here we set

$$\beta_1(t) = \begin{cases} \max\{0, -3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1, \end{cases}$$

$$\beta_2(t) = \begin{cases} 0, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 4\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1. \end{cases}$$

Then $\beta_1(t) = 0$ on $[0, 0.2] \cup [0.486, 0.771]$ and $\beta_2(t) = 0$ on $[0, 0.771]$. The distribution of $\epsilon_j \overset{iid}{\sim} N(0, 0.05)$. The number of observations in training set is n. The results of 100 Monte Carlo repetitions are showed in Table 2. This results show that the performances of estimations get better with the sample size n increase for both methods. The newly proposed method also works but the improvement is slight compared with Example 3.2.

Example 3.4: In this example, we set bivariate functional data. The coefficient functions share similar zero intervals. The difference between this example and Example 3.3 is that in this example, coefficient functions are less sparse. The motivation of this example is that we want to see whether the differences on nonzero intervals (especially the sign of coefficient functions) will influence the estimations of the new proposed method. Here we set

$$\beta_1(t) = \begin{cases} \min\{0, -3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 3\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1, \end{cases}$$

Table 3. Results of Example 3.4 obtained from 100 Monte Carlo repetitions (with standard errors in parentheses).

σ^2	n	Method	CI (%)	CZ (%)	CN (%)	ISE_0 (1e−3)	ISE_1 (1e−3)	IAE_0 (1e−3)	IAE_1 (1e−3)
0.05	300	m-SLoS	91.25	81.80	97.72	0.26	6.10	6.59	77.42
			(4.89)	(11.28)	(2.55)	(0.30)	(3.02)	(5.04)	(17.10)
		slos	88.36	67.98	99.60	0.65	3.97	12.37	65.56
			(5.79)	(15.87)	(0.71)	(0.52)	(1.35)	(7.80)	(10.03)
	500	m-SLoS	93.16	86.97	97.60	0.15	1.73	2.64	25.10
			(2.85)	(6.32)	(2.84)	(0.16)	(0.90)	(2.93)	(7.82)
		slos	88.48	68.74	99.72	0.49	2.79	10.94	53.63
			(5.35)	(14.12)	(0.51)	(0.34)	(0.68)	(6.07)	(6.77)
	1000	m-SLoS	94.78	88.18	98.74	0.10	1.22	1.61	18.95
			(2.02)	(4.80)	(2.07)	(0.11)	(0.76)	(1.48)	(6.84)
		slos	94.08	83.04	99.81	0.18	2.00	5.03	43.48
			(1.46)	(4.48)	(0.44)	(0.14)	(0.48)	(1.78)	(4.63)
0.1	300	m-SLoS	87.68	72.56	97.15	0.98	13.70	13.80	125.40
			(6.90)	(18.76)	(2.87)	(1.15)	(5.73)	(12.07)	(25.02)
		slos	82.91	53.63	98.99	2.53	13.15	28.04	121.76
			(7.64)	(20.02)	(1.56)	(2.19)	(4.32)	(20.01)	(19.28)
	500	m-SLoS	88.86	76.67	97.37	0.57	4.04	7.61	44.86
			(5.74)	(14.27)	(2.99)	(0.74)	(1.59)	(8.87)	(11.37)
		slos	84.14	57.18	99.34	1.64	8.71	22.44	98.07
			(6.31)	(16.97)	(1.07)	(1.18)	(3.25)	(12.81)	(16.3)
	1000	m-SLoS	91.43	80.73	98.34	0.22	2.55	3.53	32.76
			(4.19)	(10.52)	(2.50)	(0.25)	(1.29)	(3.51)	(10.47)
		slos	89.35	70.94	99.53	0.64	5.12	11.38	73.65
			(4.05)	(11.22)	(0.87)	(0.45)	(2.02)	(5.60)	(12.47)

$$\beta_2(t) = \begin{cases} |4\log(t+1)\sin(3.5\pi(t-0.2))|, & 0 \leqslant t \leqslant 0.486, \\ \max\{0, 4\log(t+1)\sin(3.5\pi(t-0.2))\}, & 0.486 < t \leqslant 1. \end{cases}$$

Then $\beta_1(t) = 0$ on $[0, 0.2] \cup [0.486, 0.771]$ and $\beta_2(t) = 0$ on $[0.486, 0.771]$. β_js' zero intervals are similar but not same. We set different errors to see the performances of SLoS and m-SLoS under different levels of error. The distribution of ϵ_j is $N(0, \sigma^2)$. The number of observations in the training set is n. The results of 100 Monte Carlo repetitions are showed in Table 3. The performances of estimations to both methods get better with the sample size n increase. The performances are better when the variance of ϵ_j is smaller. The improvement of the estimation of CI is slight. It is not surprising since that the zero intervals are not same for those two coefficients.

From Tables 1– 3 we can see that m-SLoS performs better than SLoS. Though CN sometimes is a little smaller when we use m-SLoS, it performs better on zero intervals. M-SLoS get a better estimation on zero intervals at the cost of sacrificing the estimation on nonzero intervals. The cost is infinitesimal that we can ignore it.

4. Application

We applied the proposed method to Tecator data. The Tecator data are recorded by a Tecator near-infrared spectrometer (the Tecator Infratec Food and Feed Analyzer) which measures the spectrum of light transmitted through a sample of minced pork meat in the region 850–1050 nm. Each sample contains finely chopped pure meat with different moisture, fat and protein contents. For each meat sample, the data consist of a 100

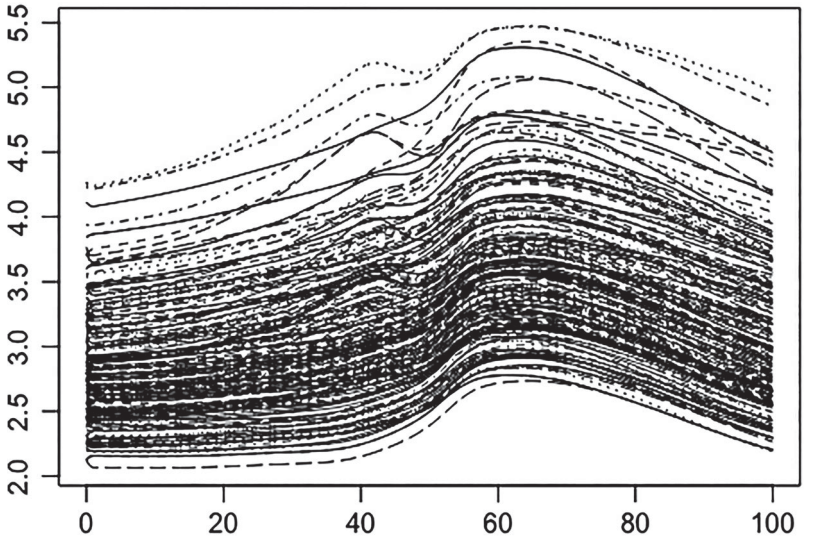

Figure 2. 100 channel spectrum of absorbances for 215 curves.

channel spectrum of absorbances and the contents of moisture (water), fat and protein. The three contents, measured in percent, are determined by analytic chemistry. The total number of samples is 215. Figure 2 displays the 215 curves. These data can be found at http://lib.stat.cmu.edu/datasets/tecator. In this section, our aim is to predict the percentage of fat and protein content given the corresponding spectrometric curve.

Fat and protein are all heat nutrients that can be converted into energy for the body to use. There must exist interconnections between those two responses. Hence when we predict fat and protein, we assume fat and protein are tightly connected. Under this assumption, we can use multivariable regressions to predict fat content or protein content.

Based on the above discussion, there must exist a range which has no prediction power either on fat content or protein content. Thus we can use m-SLoS to predict fat and protein content and investigate what range of spectra that have no predicting power on fat content and protein content. It will save energy, time and money as there is no need to record spectra on the range without prediction power.

To show the estimations and predictions of the newly proposed method comparing to SLoS, we randomly choose 170 samples from data as the training set and the remaining 45 samples are used as the testing set Q. Without loss of generality, we random split 100 times to see the difference among methods. The regularization parameters are selected by fivefold cross-validation base on the training set. Given the regularization parameters that we chose in this manner, we obtain the final estimations of the regression coefficients base on the training test and calculate the ARSE (Average Relative Square Error) on the testing set Q. Here

$$\mathrm{ARSE} = \frac{1}{45}\sum_{i\in Q}\left[\left(y_i - \hat{y}_i\right)/y_i\right]^2,$$

where y_i is the true content and $\hat{y}_i$ is the prediction. From figure 3, we can see that using m-SLoS can greatly improve the prediction both on fat and protein.

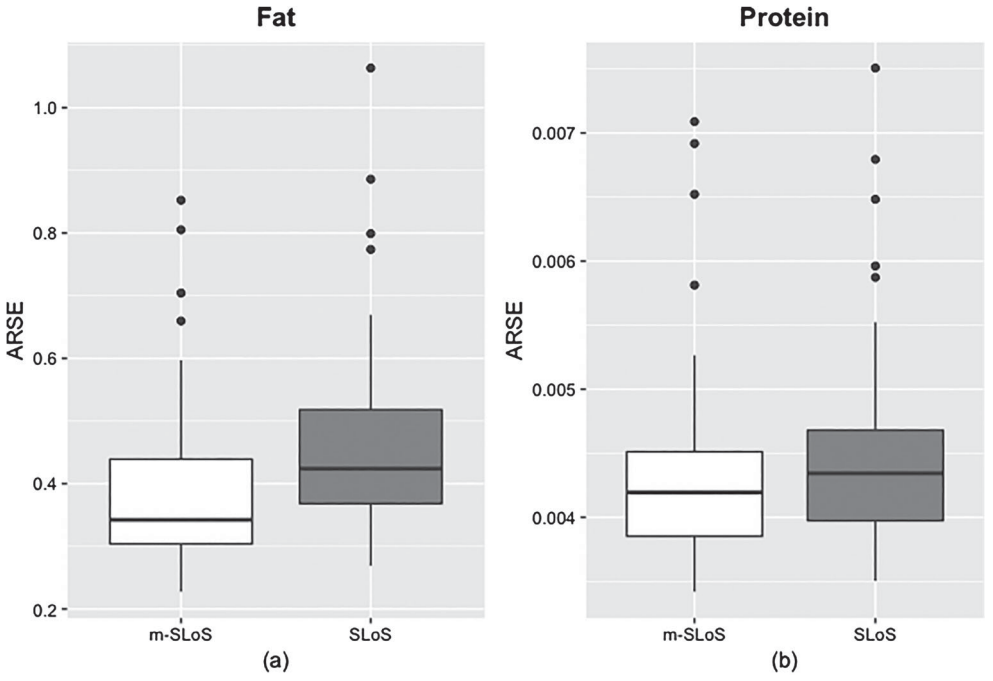

Figure 3. Comparison of two methods in terms of ARSE. (a) Comparison of SLoS and m-SLoS in terms of ARSE of prediction of fat. (b) Comparison of SLoS and m-SLoS in terms of ARSE of prediction of protein. These results were averaged over 100 random partitions of the data. The box in each box plot shows the lower quartile, median, and upper quartile values, and the whiskers show the range of ARSE in the 100 random partitions of the data.

5. Discussion

Although the literature on FLR is abundant, little has been done on interpretability and locally sparse modelling. In this article, we consider a combination of the SLoS and Laplacian quadratic as the penalty function. We call the proposed method 'm-SLoS', which uses the SLoS for encouraging locally sparse and Laplacian quadratic penalty for promoting similar local sparsity among coefficient functions associated with the interconnections among the multivariate responses. Simulations and data analysis show the excellent numerical performance of the proposed method. We have focused on the least squares based loss and functional linear regression model. Extensions to other models and robust techniques are of interest in future study.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors gratefully acknowledge National Natural Science Foundation of China (11971404, 71471152), Humanity and Social Science Youth Foundation of Ministry of Education of China (19YJC910010), Fundamental Research Funds for the Central Universities (20720171095, 20720181003) and National Institutes of Health (CA216017).

References

[1] Cuevas A, Febrero M, Fraiman R. Linear functional regression: the case of fixed design and functional response. Canad J Stat. 2002;30(2):285–300.
[2] Cardot H, Ferraty F, Mas A. Testing hypotheses in the functional linear model. Scand J Stat. 2003;30(1):241–255.
[3] Yao F, Müller HG, Wang JL. Functional data analysis for sparse longitudinal data. J Am Stat Assoc. 2005;100(470):577–590.
[4] Yao F, Müller HG, Wang JL. Functional linear regression analysis for longitudinal data. Ann Stat. 2005;33(6):2873–2903.
[5] Müller HG, Stadtmüller U. Generalized functional linear models. Ann Stat. 2005;33(2):774–805.
[6] Ramsay J, Silverman B. Functional data analysis. New York (NY): Springer-Verlag; 2005.
[7] Li Y, Hsing T. On rates of convergence in functional linear regression. J Multivar Anal. 2007;98(9):1782–1804.
[8] Tu CY, Song D, Breidt FJ, et al. Functional model selection for sparse binary time series with multiple inputs. Econ Time Ser Model Season. 2012:477–497.
[9] Wang H, Kai B. Functional sparsity: global versus local. Stat Sin. 2015;25:1337–1354.
[10] James GM, Wang J, Zhu J, et al. Functional linear regression that's interpretable. Ann Stat. 2009;37(5A):2083–2108.
[11] Zhou J, Wang NY, Wang N. Functional linear model with zero-value coefficient function at sub-regions. Stat Sin. 2013;23(1):25–50.
[12] Lin Z, Cao J, Wang L, et al. Locally sparse estimator for functional linear regression models. J Comput Graph Stat. 2016;26(2):306–318.
[13] Breiman L, Friedman JH. Predicting multivariate responses in multiple linear regression. J R Stat Soc Ser B. 1997;59(1):3–54.
[14] Rothman AJ, Levina E, Zhu J. Sparse multivariate regression with covariance estimation. J Comput Graph Stat. 2010;19(4):947–962.
[15] Peng J, Zhu J, Bergamaschi A, et al. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. Ann Appl Stat. 2010;4(1):53–77.
[16] Rai P, Kumar A, Daume H. Simultaneously leveraging output and task structures for multiple-output regression. Advances in Neural Information Processing Systems; 2012. p. 3194–3202.
[17] Bradley SP, Ben S. A cluster elastic net for multivariate regression. J Mach Learn Res. 2018;18:1–39.
[18] Shi X, Jiao Y, Yang Y, et al. Vimco: variational inference for multiple correlated outcomes in genome-wide association studies. Bioinformatics. 2019. doi:10.1093/bioinformatics/btz167.
[19] Huang J, Ma S, Li H, et al. The sparse Laplacian shrinkage estimator for high-dimensional regression. Ann Stat. 2011;39(4):2021–2046.
[20] Shi X, Zhao Q, Huang J, et al. Deciphering the associations between gene expression and copy number alteration using a sparse double Laplacian shrinkage approach. Bioinformatics. 2015;31(24):3977–3983.
[21] Wu C, Zhang Q, Jiang Y, et al. Robust network-based analysis of the associations between (epi) genetic measurements. J Multivar Anal. 2018;168:119–130.