Computational Statistics
https://doi.org/10.1007/s00180-022-01223-6

ORIGINAL PAPER

Adaptive smoothing spline estimator for the function-on-function linear regression model

Fabio Centofanti[1] · Antonio Lepore[1] · Alessandra Menafoglio[2] · Biagio Palumbo[1] · Simone Vantini[2]

Received: 23 April 2021 / Accepted: 24 March 2022

Abstract
In this paper, we propose an adaptive smoothing spline (AdaSS) estimator for the function-on-function linear regression model where each value of the response, at any domain point, depends on the full trajectory of the predictor. The AdaSS estimator is obtained by the optimization of an objective function with two spatially adaptive penalties, based on initial estimates of the partial derivatives of the regression coefficient function. This allows the proposed estimator to adapt more easily to the true coefficient function over regions of large curvature and not to be undersmoothed over the remaining part of the domain. A novel evolutionary algorithm is developed ad hoc to obtain the optimization tuning parameters. Extensive Monte Carlo simulations have been carried out to compare the AdaSS estimator with competitors that have already appeared in the literature before. The results show that our proposal mostly outperforms the competitor in terms of estimation and prediction accuracy. Lastly, those advantages are illustrated also in two real-data benchmark examples. The AdaSS estimator is implemented in the R package adass, openly available online on CRAN.

Keywords Functional data analysis · Function-on-function linear regression · Adaptive smoothing · Functional regression

JEL Classification C13 · C19

1 Introduction

Complex datasets are increasingly available due to advancements in technology and computational power and have stimulated significant methodological developments.

✉ Fabio Centofanti
fabio.centofanti@unina.it

1 Department of Industrial Engineering, University of Naples Federico II, Piazzale Tecchio 80, 80125 Naples, Italy

2 MOX - Modelling and Scientific Computing, Department of Mathematics, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy

Published online: 29 April 2022

In this regard, functional data analysis (FDA) addresses the issue of dealing with data that can be modeled as functions defined on a compact domain. FDA is a thriving area of statistics and, for a comprehensive overview, the reader could refer to Ramsay and Silverman (2005); Hsing and Eubank (2015); Horváth and Kokoszka (2012); Kokoszka and Reimherr (2017); Ferraty and Vieu (2006). In particular, the generalization of the classical multivariate regression analysis to the case where the predictor and/or the response have a functional form is referred to as functional regression and is illustrated e.g., in Morris (2015) and Ramsay and Silverman (2005). Most of the functional regression methods have been developed for models with scalar response and functional predictors (scalar-on-function regression) or functional response and scalar predictors (function-on-scalar regression). Some results may be found in Cardot et al. (2003); James (2002); Yao and Müller (2010); Müller and Stadtmüller (2005); Scheipl et al. (2015); Ivanescu et al. (2015); Hullait et al. (2021); Palumbo et al. (2020); Centofanti et al. (2020); Capezza et al. (2022). Models where both the response and the predictor are functions, namely function-on-function (FoF) regression, have been far less studied until now. In this work, we study FoF linear regression models, where the response variable function, at any domain point, depends linearly on the full trajectory of the predictor. That is,

$$Y_i\left(t\right) = \int_{\mathcal{S}} X_i\left(s\right)\beta\left(s,t\right)ds + \varepsilon_i\left(t\right) \quad t \in \mathcal{T}, \tag{1}$$

for $i = 1, \ldots, n$. The pairs (X_i, Y_i) are independent realizations of the predictor X and the response Y, which are assumed to be smooth random processes with realizations in $L^2(\mathcal{S})$ and $L^2(\mathcal{T})$, i.e., the Hilbert spaces of square integrable functions defined on the compact sets $\mathcal{S}$ and $\mathcal{T}$, respectively. Without loss of generality, the latter are also assumed with a functional mean equal to zero. The functions ε_i are zero-mean random errors, independent of X_i. The function β is smooth in $L^2(\mathcal{S} \times \mathcal{T})$, i.e., the Hilbert space of bivariate square integrable functions defined on the closed intervals $\mathcal{S} \times \mathcal{T}$, and is hereinafter referred to as *coefficient function*. For each $t \in \mathcal{T}$, the contribution of $X_i\left(\cdot\right)$ to the conditional value of $Y_i\left(t\right)$ is generated by $\beta\left(\cdot, t\right)$, which works as a continuous set of weights of the predictor evaluations. Different methods to estimate β in (1) have been proposed in the literature. Ramsay and Silverman (2005) assume the estimator of β to be in a finite dimension tensor space spanned by two basis sets and where regularization is achieved by either truncation or roughness penalties. (The latter is the foundation of the method proposed in this article as we will see below.) Yao et al. (2005b) assume the estimator of β to be in a tensor product space generated by the eigenfunctions of the covariance functions of the predictor X and the response Y, estimated by using the principal analysis by conditional expectation (PACE) method (Yao et al. 2005a). More recently, Luo and Qi (2017) propose an estimation method of the FoF linear model with multiple functional predictors based on a finite-dimensional approximation of the mean response obtained by solving a penalized generalized functional eigenvalue problem. Qi and Luo (2018) generalize the method in Luo and Qi (2017) to the high dimensional case, where the number of covariates is much larger than the sample size (i.e., $p >> n$). In order to improve model flexibility and prediction accuracy, Luo and Qi (2019) consider a FoF regression model with interaction and quadratic effects. A nonlinear FoF additive regression model with multiple functional predictors is proposed by Qi and Luo (2019).

One of the most used estimation methods is the *smoothing spline estimator* $\hat{\beta}_{SS}$ introduced by Ramsay and Silverman (2005). It is obtained as the solution of the following optimization problem

$$\begin{aligned} \hat{\beta}_{SS} &= \operatorname{argmin}_{\alpha \in \mathbb{S}_{k_1,k_2,M_1,M_2}} \left\{ \sum_{i=1}^{n} \int_{\mathcal{T}} \left[Y_i(t) - \int_{\mathcal{S}} X_i(s)\,\alpha(s,t)\,ds \right]^2 dt \right. \\ &\quad \left. + \lambda_s \int_{\mathcal{S}} \int_{\mathcal{T}} \left(\mathcal{L}_s^{m_s} \alpha(s,t) \right)^2 dsdt + \lambda_t \int_{\mathcal{S}} \int_{\mathcal{T}} \left(\mathcal{L}_t^{m_t} \alpha(s,t) \right)^2 dsdt \right\}, \end{aligned} \tag{2}$$

where $\mathbb{S}_{k_1,k_2,M_1,M_2}$ is the tensor product space generated by the sets of B-splines of orders k_1 and k_2 associated with the non-decreasing sequences of $M_1 + 2$ and $M_2 + 2$ knots defined on $\mathcal{S}$ and $\mathcal{T}$, respectively. The operators $\mathcal{L}_s^{m_s}$ and $\mathcal{L}_t^{m_t}$, with $m_s \leq k_1 - 1$ and $m_t \leq k_2 - 1$, are the m_sth and m_tth order linear differential operators applied to α with respect to the variables s and t, respectively. The two penalty terms on the right-hand side of (2) measure the roughness of the function α. The positive constants λ_s and λ_t are generally referred to as *roughness parameters* and trade off smoothness and goodness of fit of the estimator. The higher their values, the smoother the estimator of the coefficient function.

Note that the two penalty terms on the right-side hand of (2) do not depend on s and t. Therefore, the estimator $\hat{\beta}_{SS}$ may suffer from over and under smoothing when, for instance, the true coefficient function β is wiggly or peaked only in some parts of the domain. To solve this problem, we consider two adaptive roughness parameters that are allowed to vary on the domain $\mathcal{S} \times \mathcal{T}$. In this way, more flexible estimators can be obtained to improve the estimation of the coefficient function.

Methods that use adaptive roughness parameters are very popular and well established in the field of nonparametric regression and are referred to as *adaptive* methods. In particular, the smoothing spline estimator for nonparametric regression (Wahba 1990; Green and Silverman 1993; Eubank 1999; Gu 2013) has been extended by different authors to take into account the non-uniform smoothness along the domain of the function to be estimated (Ruppert and Carroll 2000; Pintore et al. 2006; Storlie et al. 2010; Wang et al. 2013; Yang and Hong 2017).

In this paper, a *spatially adaptive* estimator is proposed as the solution of the following minimization problem

$$\operatorname{argmin}_{\alpha\in\mathbb{S}_{k_1,k_2,M_1,M_2}}\left\{\sum_{i=1}^{n}\int_{\mathcal{T}}\left[Y_i\left(t\right)-\int_{\mathcal{S}}X_i\left(s\right)\alpha\left(s,t\right)ds\right]^2dt\right.$$
$$\left.+\int_{\mathcal{S}}\int_{\mathcal{T}}\lambda_s\left(s,t\right)\left(\mathcal{L}_s^{m_s}\alpha\left(s,t\right)\right)^2dsdt+\int_{\mathcal{S}}\int_{\mathcal{T}}\lambda_t\left(s,t\right)\left(\mathcal{L}_t^{m_t}\alpha\left(s,t\right)\right)^2dsdt\right\},\quad(3)$$

where the two roughness parameters $\lambda_s\left(s,t\right)$ and $\lambda_t\left(s,t\right)$ are functions that produce different amount of penalty, and, thus, allow the estimator to spatially adapt, i.e., to accommodate varying degrees of roughness over the domain $\mathcal{S}\times\mathcal{T}$. Therefore, the model may accommodate the local behavior of β by imposing a heavier penalty in regions of lower smoothness. Because $\lambda_s\left(s,t\right)$ and $\lambda_t\left(s,t\right)$ are intrinsically infinite dimensional, their specification could be rather complicated without further assumptions.

The proposed estimator is applied to FoF linear regression model reported in (1), and is referred to as adaptive smoothing spline (AdaSS) estimator. It is obtained as the solution of the optimization problem in (3), with $\lambda_s\left(s,t\right)$ and $\lambda_t\left(s,t\right)$ chosen based on an initial estimate of the partial derivatives $\mathcal{L}_s^{m_s}\alpha\left(s,t\right)$ and $\mathcal{L}_t^{m_t}\alpha\left(s,t\right)$. The rationale behind this choice is to allow the contribution of $\lambda_s\left(s,t\right)$ and $\lambda_t\left(s,t\right)$, to the penalties in (3), to be small over regions where the initial estimate has large m_sth and m_tth curvatures (i.e., partial derivatives), respectively. This can be regarded as an extension to the FoF linear regression model of the idea of Storlie et al. (2010) and Abramovich and Steinberg (1996). Moreover, to overcome some limitations of the most famous grid-search method (Bergstra et al. 2011), a new evolutionary algorithm is proposed for the choice of the unknown parameters, needed to compute the AdaSS estimator. The method presented in this article is implemented in the R package `adass`, openly available online on CRAN.

The rest of the paper is organized as follows. In Sect. 2.1, the proposed estimator is presented. Computational issues involved in the AdaSS estimator calculation are discussed in Sects. 2.2 and 2.3. In Sect. 3, by means of a Monte Carlo simulation study, the performance of the proposed estimator is compared with those achieved by competing estimators already appeared in the literature. Lastly, two real-data examples are presented in Sect. 4 to illustrate the practical applicability of the proposed estimator. The conclusion is in Sect. 5. Supplementary Material is available online where the derivation of the approximations of the AdaSS penalty (Supplementary Material 1) and additional simulation studies are presented (Supplementary Material 2) together with the optimal tuning parameter selected in the simulation study in Sect. 3 (Supplementary Material 3).

2 The adaptive smoothing spline estimator

2.1 The estimator

The AdaSS estimator $\hat{\beta}_{AdaSS}$ is defined as the solution of the optimization problem in (3) where the two roughness parameters $\lambda_s(s,t)$ and $\lambda_t(s,t)$ are as follows

$$\lambda_s(s,t) = \lambda_s^{AdaSS} \frac{1}{\left(|\widehat{\beta_s^{m_s}}(s,t)| + \delta_s\right)^{\gamma_s}}, \quad \lambda_t(s,t) = \lambda_t^{AdaSS} \frac{1}{\left(|\widehat{\beta_t^{m_t}}(s,t)| + \delta_t\right)^{\gamma_t}}$$

that is,

$$\begin{aligned} \hat{\beta}_{AdaSS} &= \operatorname{argmin}_{\alpha \in \mathbb{S}_{k_1,k_2,M_1,M_2}} \Bigg\{ \sum_{i=1}^{n} \int_{\mathcal{T}} \left[Y_i(t) - \int_{\mathcal{S}} X_i(s)\alpha(s,t)\,ds \right]^2 dt \\ &\quad + \lambda_s^{AdaSS} \int_{\mathcal{S}} \int_{\mathcal{T}} \frac{1}{\left(|\widehat{\beta_s^{m_s}}(s,t)| + \delta_s\right)^{\gamma_s}} \left(\mathcal{L}_s^{m_s}\alpha(s,t)\right)^2 ds dt \\ &\quad + \lambda_t^{AdaSS} \int_{\mathcal{S}} \int_{\mathcal{T}} \frac{1}{\left(|\widehat{\beta_t^{m_t}}(s,t)| + \delta_t\right)^{\gamma_t}} \left(\mathcal{L}_t^{m_t}\alpha(s,t)\right)^2 ds dt \Bigg\}, \end{aligned} \tag{4}$$

for some tuning parameters $\lambda_s^{AdaSS}, \delta_s, \gamma_s, \lambda_t^{AdaSS}, \delta_t, \gamma_t \geq 0$ and $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ initial estimates of $\mathcal{L}_s^{m_s}\beta$ and $\mathcal{L}_t^{m_t}\beta$, respectively. Note that the two roughness parameters λ_s and λ_t assume large values over domain regions where $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ are small. Therefore, in the right-hand side of (4), $(\mathcal{L}_s^{m_s}\alpha)^2$ and $(\mathcal{L}_t^{m_t}\alpha)^2$ are weighted through the inverse of $|\widehat{\beta_s^{m_s}}|$ and $|\widehat{\beta_t^{m_t}}|$. That is, over domain regions where $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ are small, $(\mathcal{L}_s^{m_s}\alpha)^2$ and $(\mathcal{L}_t^{m_t}\alpha)^2$ have larger weights than over those regions where $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ are large. For this reasons, the final estimator is able to adapt to the coefficient function over regions of large curvature without over smoothing it over regions where the m_sth and m_tth curvatures are small.

The constants δ_s and δ_t allow $\hat{\beta}_{AdaSS}$ not to have m_sth and m_tth-order inflection points at the same location of $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$, respectively. Indeed, when δ_s and δ_t are set to zero, where $\widehat{\beta_s^{m_s}} = 0$ and $\widehat{\beta_t^{m_t}} = 0$ (m_sth and m_tth-order inflection points), the corresponding penalties go to infinite, and, thus, $\mathcal{L}_s^{m_s}\alpha(s,t)$ and $\mathcal{L}_t^{m_t}\alpha(s,t)$ become zero in accordance with the minimization problem. Therefore, the presence of δ_s and δ_t makes $\hat{\beta}_{AdaSS}$ more robust against the choice of the initial estimate of the linear differential operators applied to β with respect to s and t. Finally, γ_s and γ_t control the amount of weight placed in $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$, whereas λ_s^{AdaSS} and λ_t^{AdaSS} are smoothing parameters. The solution of the optimization problem in (4) can be obtained in a closed form if the penalty terms are approximated as described in Sect. 2.2. There are several choices for the initial estimates $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$. As in Abramovich and Steinberg

(1996), we suggest to apply the m_sth and m_tth order linear differential operator to the smoothing spline estimator $\hat{\beta}_{SS}$ in (2).

2.2 The derivation of the AdaSS estimator

The minimization in (4) is carried out over $\alpha \in \mathbb{S}_{k_1,k_2,M_1,M_2}$. This implicitly means that we are approximating β as follows

$$\beta(s,t) \approx \tilde{\beta}(s,t) = \sum_{i=1}^{M_1+k_1} \sum_{j=1}^{M_2+k_2} b_{ij} \psi_i^s(s) \psi_j^t(t)$$
$$= \boldsymbol{\psi}^s(s)^T \boldsymbol{B} \boldsymbol{\psi}^t(t) \quad s \in \mathcal{S}, t \in \mathcal{T}, \tag{5}$$

where $\boldsymbol{B} = \{b_{ij}\} \in \mathbb{R}^{(M_1+k_1)\times(M_2+k_2)}$. The two sets $\boldsymbol{\psi}^s = \left(\psi_1^s, \ldots, \psi_{M_1+k_1}^s\right)^T$ and $\boldsymbol{\psi}^t = \left(\psi_1^t, \ldots, \psi_{M_2+k_2}^t\right)^T$ are B-spline functions of order k_1 and k_2 and non-decreasing knots sequences $\Delta^s = \{s_0, s_1, \ldots, s_{M_1}, s_{M_1+1}\}$ and $\Delta^t = \{t_0, t_1, \ldots, t_{M_2}, t_{M_2+1}\}$, defined on $\mathcal{S} = \left[s_0, s_{M_1+1}\right]$ and $\mathcal{T} = \left[t_0, t_{M_2+1}\right]$, respectively, that generate $\mathbb{S}_{k_1,k_2,M_1,M_2}$. Thus, estimating β in (4) means estimating $\boldsymbol{B}$. Let $\alpha(s,t) = \boldsymbol{\psi}^s(s)^T \boldsymbol{B}_\alpha \boldsymbol{\psi}^t(t)$, $s \in \mathcal{S}, t \in \mathcal{T}$, in $\mathbb{S}_{k_1,k_2,M_1,M_2}$, where $\boldsymbol{B}_\alpha = \{b_{\alpha,ij}\} \in \mathbb{R}^{(M_1+k_1)\times(M_2+k_2)}$. Then, the first term of the right-hand side of (4) may be rewritten as (see Ramsay and Silverman (2005), pag 291-293, for the derivation)

$$\sum_{i=1}^{n} \int_{\mathcal{T}} \left[Y_i(t) - \int_{\mathcal{S}} X_i(s)\alpha(s,t)\,ds\right]^2 dt$$
$$= \sum_{i=1}^{n} \int_{\mathcal{T}} Y_i(t)^2\,dt - 2\,\mathrm{Tr}\left[\boldsymbol{X}\boldsymbol{B}_\alpha \boldsymbol{Y}^T\right] + \mathrm{Tr}\left[\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{B}_\alpha \boldsymbol{W}_t \boldsymbol{B}_\alpha^T\right], \tag{6}$$

where $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)^T$, with $\boldsymbol{X}_i = \int_{\mathcal{S}} X_i(s)\,\boldsymbol{\psi}^s(s)\,ds$, $\boldsymbol{Y} = (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n)^T$ with $\boldsymbol{Y}_i = \int_{\mathcal{T}} Y_i(t)\,\boldsymbol{\psi}^t(t)\,dt$, and $\boldsymbol{W}_t = \int_{\mathcal{T}} \boldsymbol{\psi}^t(t)\,\boldsymbol{\psi}^t(t)^T\,dt$. The term $\mathrm{Tr}[\boldsymbol{A}]$ denotes the trace of a square matrix $\boldsymbol{A}$.

In order to simplify the integrals in the two penalty terms on the right-hand side of (4), and thus obtain a linear form in $\boldsymbol{B}_\alpha$, we consider, for $s \in \mathcal{S}$ and $t \in \mathcal{T}$, the following approximations of $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$

$$\widehat{\beta_s^{m_s}}(s,t) \approx \sum_{i=0}^{L_s} \sum_{j=0}^{L_t} \widehat{\beta_s^{m_s}}\left(\tau_{s,i+1}, \tau_{t,j+1}\right) I_{[(\tau_{s,i},\tau_{s,i+1})\times(\tau_{t,j},\tau_{t,j+1})]}(s,t), \tag{7}$$

and

$$\widehat{\beta_t^{m_t}}(s,t) \approx \sum_{i=0}^{L_s} \sum_{j=0}^{L_t} \widehat{\beta_t^{m_t}}\left(\tau_{s,i+1}, \tau_{t,j+1}\right) I_{[(\tau_{s,i},\tau_{s,i+1})\times(\tau_{t,j},\tau_{t,j+1})]}(s,t), \tag{8}$$

where $\Theta^s = \{\tau_{s,0}, \tau_{s,1}, \dots \tau_{s,L_s}, \tau_{s,L_s+1}\}$ and $\Theta^t = \{\tau_{t,0}, \tau_{t,1}, \dots \tau_{t,L_t}, \tau_{t,L_t+1}\}$ are non decreasing knot sequences with $\tau_{s,0} = s_0$, $\tau_{s,L_s+1} = s_{M_1+1}$, $\tau_{t,0} = t_0$, $\tau_{t,L_t+1} = t_{M_2+1}$, and $I_{[a \times b]}(z_1, z_2) = 1$ for $(z_1, z_2) \in [a \times b]$ and zero elsewhere. In (7) and (8), we are assuming that $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ are well approximated by a piecewise constant function, whose values are constant on rectangles defined by the two knot sequences Θ^s and Θ^t. It can be easily proved, by following Schumaker Schumaker (2007) (pag. 491, Theorem 12.7), that the approximation error in both cases goes to zero as the mesh widths $\overline{\delta}^s = \max_i \left(\tau_{s,i+1} - \tau_{s,i}\right)$ and $\overline{\delta}^t = \max_j \left(\tau_{t,j+1} - \tau_{t,j}\right)$ go to zero. Therefore, $\widehat{\beta_s^{m_s}}$ and $\widehat{\beta_t^{m_t}}$ can be exactly recovered by uniformly increasing the number of knots L_s and L_t. In this way, the two penalties on the right-hand side of (4) can be rewritten as (Supplementary Material 1)

$$\lambda_s^{AdaSS} \int_{\mathcal{S}} \int_{\mathcal{T}} \frac{1}{\left(|\widehat{\beta_s^{m_s}}(s,t)| + \delta_s\right)^{\gamma_s}} \left(\mathcal{L}_s^{m_s} \alpha(s,t)\right)^2 ds dt$$
$$\approx \lambda_s^{AdaSS} \sum_{i=1}^{L_s+1} \sum_{j=1}^{L_t+1} d_{ij}^s \operatorname{Tr}\left[\boldsymbol{B}_\alpha^T \boldsymbol{R}_{s,i} \boldsymbol{B}_\alpha \boldsymbol{W}_{t,j}\right] \tag{9}$$

and

$$\lambda_t^{AdaSS} \int_{\mathcal{S}} \int_{\mathcal{T}} \frac{1}{\left(|\widehat{\beta_t^{m_t}}(s,t)| + \delta_t\right)^{\gamma_t}} \left(\mathcal{L}_t^{m_t} \alpha(s,t)\right)^2 ds dt$$
$$\approx \lambda_t^{AdaSS} \sum_{i=1}^{L_s+1} \sum_{j=1}^{L_t+1} d_{ij}^t \operatorname{Tr}\left[\boldsymbol{B}_\alpha^T \boldsymbol{W}_{s,i} \boldsymbol{B}_\alpha \boldsymbol{R}_{t,j}\right], \tag{10}$$

where $\boldsymbol{W}_{s,i} = \int_{[\tau_{s,i-1}, \tau_{s,i}]} \boldsymbol{\psi}^s(s) \boldsymbol{\psi}^s(s)^T ds$, $\boldsymbol{W}_{t,j} = \int_{[\tau_{t,j-1}, \tau_{t,j}]} \boldsymbol{\psi}^t(t) \boldsymbol{\psi}^t(t)^T dt$, $\boldsymbol{R}_{s,i} = \int_{[\tau_{s,i-1}, \tau_{s,i}]} \mathcal{L}_s^{m_s}\left[\boldsymbol{\psi}^s(s)\right] \mathcal{L}_s^{m_s}\left[\boldsymbol{\psi}^s(s)\right]^T ds$ and $\boldsymbol{R}_{t,j} = \int_{[\tau_{t,j-1}, \tau_{t,j}]} \mathcal{L}_t^{m_t}\left[\boldsymbol{\psi}^t(t)\right] \mathcal{L}_t^{m_t}\left[\boldsymbol{\psi}^t(t)\right]^T dt$, and $d_{ij}^s = \left\{\frac{1}{\left(|\widehat{\beta_s^{m_s}}(\tau_{s,i}, \tau_{t,j})| + \delta_s\right)^{\gamma_s}}\right\}$ and $d_{ij}^t = \left\{\frac{1}{\left(|\widehat{\beta_t^{m_t}}(\tau_{s,i}, \tau_{t,j})| + \delta_t\right)^{\gamma_t}}\right\}$, for $i = 1, \dots, L_s + 1$ and $j = 1, \dots, L_t + 1$.

The optimization problem in (4) can be then approximated with the following

$$\hat{\boldsymbol{B}}_{AS} \approx \operatorname{argmin}_{\boldsymbol{B}_\alpha \in \mathbb{R}^{(M_1+k_1)\times(M_2+k_2)}} \left\{ \sum_{i=1}^{n} \int_{\mathcal{T}} Y_i(t)^2 dt - 2 \operatorname{Tr}\left[\boldsymbol{X} \boldsymbol{B}_\alpha \boldsymbol{Y}^T\right] + \operatorname{Tr}\left[\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{B}_\alpha \boldsymbol{W}_t \boldsymbol{B}_\alpha^T\right] \right.$$
$$\left. + \sum_{i=1}^{L_s+1} \sum_{j=1}^{L_t+1} \left(\lambda_s^{AdaSS} d_{ij}^s \operatorname{Tr}\left[\boldsymbol{B}_\alpha^T \boldsymbol{R}_{s,i} \boldsymbol{B}_\alpha \boldsymbol{W}_{t,j}\right] + \lambda_t^{AdaSS} d_{ij}^t \operatorname{Tr}\left[\boldsymbol{B}_\alpha^T \boldsymbol{W}_{s,i} \boldsymbol{B}_\alpha \boldsymbol{R}_{t,j}\right]\right) \right\}, \tag{11}$$

or by vectorization as

$$\hat{\boldsymbol{b}}_{AS} \approx \operatorname{argmin}_{\boldsymbol{b}_\alpha \in \mathbb{R}^{(M_1+k_1)(M_2+k_2)}} \Big\{ -2\operatorname{vec}\left(\boldsymbol{X}^T\boldsymbol{Y}\right)^T \boldsymbol{b}_\alpha + \boldsymbol{b}_\alpha^T \left(\boldsymbol{W}_t \otimes \boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{b}_\alpha$$
$$\sum_{i=1}^{L_s+1}\sum_{j=1}^{L_t+1}\left(\lambda_s^{AdaSS} d_{ij}^s \boldsymbol{b}_\alpha^T \boldsymbol{L}_{wr,ij}\boldsymbol{b}_\alpha + \lambda_t^{AdaSS} d_{ij}^t \boldsymbol{b}_\alpha^T \boldsymbol{L}_{rw,ij}\boldsymbol{b}_\alpha\right)\Big\}, \tag{12}$$

where $\hat{\boldsymbol{b}}_{AS} = \operatorname{vec}\left(\hat{\boldsymbol{B}}_{AS}\right)$, $\boldsymbol{L}_{rw,ij} = \left(\boldsymbol{R}_{t,j} \otimes \boldsymbol{W}_{s,i}\right)$ and $\boldsymbol{L}_{wr,ij} = \left(\boldsymbol{W}_{t,j} \otimes \boldsymbol{R}_{s,i}\right)$, for $i = 1, \ldots, L_s + 1$ and $j = 1, \ldots, L_t + 1$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{j\times k}$, $\operatorname{vec}(\boldsymbol{A})$ indicates the vector of length jk obtained by writing the matrix $\boldsymbol{A}$ as a vector column-wise, and $\otimes$ is the Kronecker product. Then, the minimizer of the optimization problem in (12) has the following expression

$$\begin{aligned}
&\hat{\boldsymbol{b}}_{AdaSS}\\
&\approx \left[\left(\boldsymbol{W}_t \otimes \boldsymbol{X}^T\boldsymbol{X}\right) + \sum_{i=1}^{L_s+1}\sum_{j=1}^{L_t+1}\left(\lambda_s^{AdaSS} d_{ij}^s \boldsymbol{L}_{wr,ij} + \lambda_t^{AdaSS} d_{ij}^t \boldsymbol{L}_{rw,ij}\right)\right]^{-1} \operatorname{vec}\left(\boldsymbol{X}^T\boldsymbol{Y}\right)\\
&= \boldsymbol{K}^{-1}\operatorname{vec}\left(\boldsymbol{X}^T\boldsymbol{Y}\right),
\end{aligned} \tag{13}$$

where

$$\boldsymbol{K} = \left(\boldsymbol{W}_t \otimes \boldsymbol{X}^T\boldsymbol{X}\right) + \sum_{i=1}^{L_s+1}\sum_{j=1}^{L_t+1}\left(\lambda_s^{AdaSS} d_{ij}^s \boldsymbol{L}_{wr,ij} + \lambda_t^{AdaSS} d_{ij}^t \boldsymbol{L}_{rw,ij}\right).$$

The identifiability of β, i.e., the uniqueness of $\hat{\boldsymbol{b}}_{AdaSS}$, comes from the fact that the inverse of $\boldsymbol{K}$ exists. This is guaranteed with probability tending to one as the sample size increases, under the condition that the covariance operator of X is strictly positive, i.e., his kernel is empty (Prchal and Sarda, 2007). In Equation (13), this reverts into the condition that $\boldsymbol{X}^T\boldsymbol{X}$ is positive definite. Moreover, Scheipl and Greven (2016) show that identifiability still holds also in the case of rank deficiency of $\left(\boldsymbol{W}_t \otimes \boldsymbol{X}^T\boldsymbol{X}\right)$ if, and only if, the kernel of the covariate covariance operator does not overlap that of the roughness penalties.

To obtain $\hat{\boldsymbol{b}}_{AdaSS}$ in (13) the tuning parameters $\lambda_s^{AdaSS}, \delta_s, \gamma_s, \lambda_t^{AdaSS}, \delta_t, \gamma_t$ must be opportunely chosen. This issue is discussed in Sect. 2.3.

2.3 The algorithm for the parameter selection

There are some tuning parameters in the optimization problem (12) that must be chosen to obtain the AdaSS estimator. Usually, the tensor product space $\mathbb{S}_{k_1,k_2,M_1,M_2}$ is chosen with $k_1 = k_2 = 4$, i.e., cubic B-splines, and equally spaced knot sequences. Although the choice of M_1 and M_2 is not crucial (Cardot et al. 2003), it should allow the final estimator to capture the local behaviour of the coefficient function β, that is, M_1 and

M_2 should be sufficiently large. The smoothness of the final estimator is controlled by the two penalty terms on the right-hand side of (12).

The tuning parameters $\lambda_s^{AdaSS}, \delta_s, \gamma_s, \lambda_t^{AdaSS}, \delta_t, \gamma_t$ could be fixed by using the conventional K-fold cross validation (CV) (Hastie et al. 2009), where the combination of the parameters to be explored are chosen by means of the classic grid search method (Hastie et al. 2009). That is, an exhaustive searching through a manually specified subset of the tuning parameter space (Bergstra and Bengio 2012). Although, in our setting, grid search is embarrassingly parallel (Herlihy and Shavit 2011), it is not scalable because it suffers from the curse of dimensionality. However, even if this is beyond the scope of the present work, note that the number of combinations to explore grows exponentially with the number of tuning parameters and makes unsuitable the application of the proposed method to the FoF linear model in the case of multiple predictors. Then, to facilitate the use of the proposed method by practitioners, in what follows, we proposed a novel evolutionary algorithm for tuning parameter selection, referred to as *evolutionary algorithm for the adaptive smoothing spline estimator* (EAASS) inspired by the *population based training* (PBT) introduced by Jaderberg et al. (2017). The PBT algorithm was introduced to address the issue of hyperparameter optimization for neural networks. It bridges and extends parallel search method (e.g., grid search and random search) with sequential optimization method (e.g., hand tuning and Bayesian optimization). The former runs many parallel optimization processes, for different combinations of hyperparameter values, and, then chooses the combination that shows the best performance. The latter performs several steps of few parallel optimizations, where, at each step, information coming from the previous step is used to identify the combinations of hyperparameter values to explore. For further details on the PBT algorithm the readers should refer to Jaderberg et al. (2017), where the authors demonstrated its effectiveness and wide applicability. The pseudo code of the EAASS algorithm is given in Algorithm 1.

Algorithm 1 EAASS algorithm

```
Choose the initial population $\mathcal{P} = \{p_i\}$ of combinations of tuning parameter values
Obtain the set $\mathcal{V} = \{v_i\}$ of estimated prediction errors corresponding to $\mathcal{P}$
repeat
  Identify the set $\mathcal{Q} \subseteq \mathcal{P}$ and the corresponding $\mathcal{Z} \subseteq \mathcal{V}$      ▶ exploitation
  for $p_i \in \mathcal{Q}$ do                                                                      ▶ exploration
    Obtain the new combination of tuning parameter values, $p_i'$
    Obtain the new estimated prediction error $v_i'$ corresponding to $p_i'$
  end for
  Define $\mathcal{Q}' = \{p_i'\}$ and $\mathcal{Z}' = \{v_i'\}$
  Set $\mathcal{P} = \mathcal{P} \setminus \mathcal{Q} \cup \mathcal{Q}'$ and $\mathcal{V} = \mathcal{V} \setminus \mathcal{Z} \cup \mathcal{Z}'$
until The stopping condition is met
Return $p_i \in \mathcal{P}$ with the lowest $v_i \in \mathcal{V}$
```

The first step is the identification of an initial population $\mathcal{P}$ of the tuning parameter combinations p_is. This can be done, for each combination and each tuning parameter, by randomly selecting a value in a pre-specified range. Then, the set $\mathcal{V}$ of estimated prediction errors v_is corresponding to $\mathcal{P}$ is obtained by means of K-fold CV. We choose a subset $\mathcal{Q}$ of $\mathcal{P}$, by following a given exploitation strategy and, thus, the corresponding subset $\mathcal{Z}$ of $\mathcal{V}$. A typical exploitation strategy is the *truncation selection*, where the worse $r\%$, for $0 \leq r \leq 100$, of $\mathcal{P}$, in terms of estimated prediction error, is substituted by elements randomly sampled from the remaining $(100 - r)\%$ part of the current population (Jaderberg et al. 2017). Then the following step consists of an exploration strategy where the tuning parameter combinations in $\mathcal{Q}$ are substituted by new ones. The simulation study in Sect. 3 and the real-data Examples in Sect. 4 are based on a *perturbation* where each tuning parameter value of the given combination is randomly perturbed by a factor of 1.2 or 0.8. The exploitation and exploration phases are repeated until a stopping condition is met, e.g, maximum number of iterations. Other exploration and exploitation strategies can be found in Bäck et al. (1997). At last, the selected tuning parameter combination is obtained as an element of $\mathcal{P}$ that achieves the lowest estimated prediction error.

The choice of the initial population $\mathcal{P}$ may have an impact on the selection of the optimal tuning parameters. Although a perturbation step in the EAASS algorithm may allow escaping from $\mathcal{P}$, it cannot be excluded that if the optimal tuning parameter combination is too far from this region, the algorithm may terminate before reaching it. This behaviour can be mitigated by either considering an adaptive stopping condition, which depends on the prediction error improvement between two consecutive iterations, or appropriately selecting $\mathcal{P}$. However, in some cases the former solution may need too many iterations of the EAAS algorithm and consequently, it may result inefficiently slow. The latter solution is more suitable as a general recommendation and can be specifically implemented by preliminary running a few iterations of the EAASS algorithm to assess the coherence of the chosen region with the optimal tuning parameter combination. This is in fact the procedure used to select $\mathcal{P}$ in both the simulation study of Sect. 3 and real-data examples of Sect. 4. Moreover, as explained in Jaderberg et al. (2017), when the size of $\mathcal{P}$ is too small, the performance of the PBT algorithm may deteriorate. Being a greedy algorithm, the PBT algorithm may in fact get stuck in local optima for small population sizes. A simple guideline is to select the population size as large as possible to maintain enough diversity and scope for exploration with respect to the computational resources available.

The intuition on which the EAASS algorithm is based is quite straightforward. Instead of finding the optimal tuning parameter combination across the whole parameter space, which is clearly infeasible, the combinations in $\mathcal{P}$ are the only ones to be considered in two phases, i.e., the exploitation and exploration phases. The former decides the combinations in $\mathcal{P}$ that should be abandoned to focus on more promising ones, whereas the latter proposes new parameter combinations to better explore the tuning parameter space. In this way, tuning parameter combinations with unsatisfactory predictive performance are overlooked. Then, the computational effort is focused on tuning parameter regions that are closer to the tuning parameter combinations with the smallest prediction errors. Moreover, models obtained for each tuning parameter combination in $\mathcal{P}$ could be estimated in a parallel fashion. In such a way, the EAASS algorithm comprises the features of both the parallel search and sequential methods.

3 Simulation study

In this section, the performance of the AdaSS estimator is assessed on several simulated datasets. In particular, we compare the AdaSS estimator with cubic B-splines and $m_s = m_t = 2$ with five competing methods that represent the state of the art in the FoF liner regression model estimation. The first two are those proposed by Ramsay and Silverman (2005). The first one, hereinafter referred to as SMOOTH estimator, is the smoothing spline estimator described in (2), whereas, the second one, hereinafter referred to as TRU estimator, assumes that the coefficient function is in a finite dimensional tensor product space generated by two sets of B-splines with regularization achieved by choosing the space dimension. Then, we consider also the estimator proposed by Yao et al. (2005b) and Canale and Vantini (2016). The former is based on the functional principal component decomposition, and is hereinafter referred to as PCA estimator, while the latter relies on a ridge type penalization, hereinafter referred to as RIDGE estimator. Lastly, as the fifth alternative, we explore the estimator proposed by Luo and Qi (2017), hereinafter referred to as SIGCOMP. For illustrative purposes, we also consider a version of the AdaSS estimator, referred to AdaSStrue, whose roughness parameters are calculated by assuming that the true coefficient function is known. Obviously, the AdaSStrue has not a practical meaning because the true coefficient function is never known. However, it allows one to better understand the influence of the initial estimates of the partial derivatives on the AdaSS performance. All the unknown parameters of the competing methods considered are chosen by means of a 10-fold CV. The tuning parameters of the AdaSS and AdaSStrue estimators are chosen through the EAASS algorithm. The initial population $\mathcal{P}$ of tuning parameter combinations for the EAASS algorithm is generated by randomly selecting 24 values in pre-specified ranges. Specifically, δ_s and δ_t are uniformly sampled in $\left[0, 0.1 \max |\widehat{\beta_s^{m_s}}(s,t)|\right]$ and $\left[0, 0.1 \max |\widehat{\beta_t^{m_t}}(s,t)|\right]$, respectively; the constants γ_s and γ_t are uniformly sampled in $[0, 4]$; and the two roughness parameters λ_s and λ_t are uniformly selected in $\left[10^{-8}, 10^3\right]$. The set $\mathcal{V}$ is obtained by using a 10-fold CV, the exploitation and exploration phases are as described in Sect. 2.3 and a maximum number of iterations equal to 15 is set as the stopping condition. For each simulation, a training sample of n observations is generated along with a test set T of size $N = 4000$. They are used to estimate β and to test the predictive performance of the estimated model, respectively. Three different sample sizes are considered, viz., $n = 150, 500, 1000$. The estimation accuracy of the estimators are assessed by using the *integrated squared error* (ISE) defined as

$$\mathrm{ISE} = \frac{1}{A} \int_{\mathcal{S}} \int_{\mathcal{T}} \left(\hat{\beta}(s,t) - \beta(s,t)\right)^2 ds dt, \tag{14}$$

where A is the measure of $\mathcal{S} \times \mathcal{T}$. The ISE aims to measure the estimation error of $\hat{\beta}$ with respect to β. Whereas, the predictive accuracy is measured through the *prediction mean squared error* (PMSE) defined as

$$\mathrm{PMSE} = \frac{1}{N} \sum_{(X,Y) \in T} \int_{\mathcal{T}} \left(Y(t) - \int_{\mathcal{S}} X(s) \hat{\beta}(s,t)\, ds \right)^2 dt.$$

The observations in the test set are centred by subtracting from each observation the corresponding sample mean function estimated in the training set. The observations in the training and test sets are obtained as follows. The covariate X_i and the errors ε_i are generated as a linear combination of cubic B-splines, Ψ_i^x and Ψ_i^{ε}, with evenly spaced knots, i.e., $X_i = \sum_{j=1}^{32} x_{ij} \Psi_i^x$ and $\varepsilon_i = k \sum_{j=1}^{20} e_{ij} \Psi_i^{\varepsilon}$. The coefficients x_{ij} and e_{ij}, for $i = 1, \dots, n$; $j = 1, \dots, 32$ and $j = 1, \dots, 20$, are independent realizations of standard normal random variable and the numbers of basis have been randomly chosen between 10 and 50. The constant k is chosen such that the signal-to-noise ratio $SN \doteq \int_{\mathcal{T}} \mathrm{Var}_X[\mathrm{E}(Y_i|X_i)] / \int_{\mathcal{T}} \mathrm{Var}(\varepsilon_i)$ is equal to 4, where Var_X is the variance with respect to the random covariate X. Then, given the coefficient function β, the response Y_i is obtained.

It is worth remarking that the coefficient function β is not identifiable in $L^2(\mathcal{S} \times \mathcal{T})$, because X_i is generated as a finite linear combination of basis functions. This means that the null space of K_X, i.e., the covariance operator of X_i, is not empty. In this case, the coefficient function β is identifiable only in the closure of the image of K_X (Cardot et al. 2003), which is denoted by $Im(K_X) = \{K_X f : f \in L^2(\mathcal{S})\}$. Thus, to obtain a meaningful measure of the estimation accuracy, ISE should be computed by considering estimate projections onto $Im(K_X)$. This allows the estimation method performance to be compared over the identifiable part of the model, only. Also in accordance with James et al. (2009); Zhou et al. (2013); Lin et al. (2017), the space spanned by the 32 cubic B-splines used to generate X_i is sufficiently rich to reconstruct the true coefficient function β and its estimate $\hat{\beta}$ for the proposed and competing methods. Hence, ISE in Equation (14) can be still suitably used.

In the Supplementary Material 2, additional simulation studies are presented to study the performance of both the proposed estimator with respect to different choices of partial derivative estimates (Supplementary Material 2.1) and the repeated application of the AdaSS estimation method (Supplementary Material 2.2).

3.1 Mexican hat function

The Mexican hat function is a linear function with a sharp smoothness variation in central part of the domain. In this case, the coefficient function β is defined as

$$\beta(s,t) = -1 + 1.5s + 1.5t + 0.05\phi(s,t), \quad s,t \in [0,1] \times [0,1]$$

where ϕ is a multivariate normal distribution with mean $\boldsymbol{\mu} = (0.6, 0.6)^T$ and diagonal covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(0.001, 0.001)$. Figure 1 displays the AdaSS and the SMOOTH estimates along with the true coefficient function for a randomly selected simulation run.

The proposed estimator tends to be smoother in the flat region and is able to better capture the peak in the coefficient function (at $t \approx 0.6$) than the SMOOTH estimate. The latter, to perform reasonably well along the whole domain, selects tuning param-

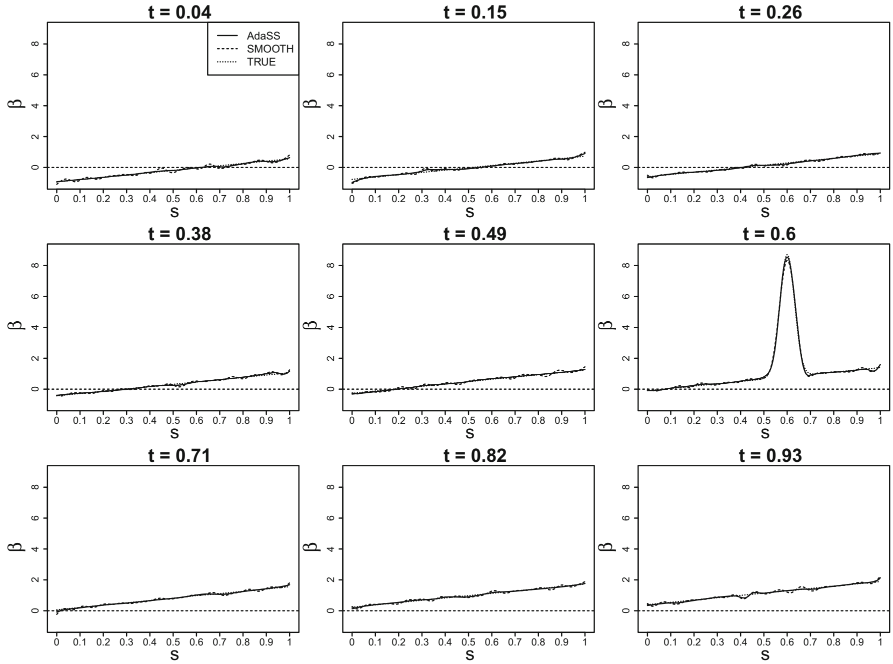

Fig. 1 AdaSS (solid line) and SMOOTH (dashed line) estimates of the coefficient functions and the TRUE coefficient function β (dotted line) for different values of t in the case of the Mexican hat function

eters that may be not sufficiently small on the peaky region, or not sufficiently large over the flat region. This is also confirmed by the graphical appeal of the AdaSS estimate with respect to the competitor ones. In Fig. 2 and top of Table 1, the values of ISE and PMSE achieved by the AdaSS, AdaSStrue, and competitor estimators are shown as functions of the sample size n. Without considering the AdaSStrue estimator, the AdaSS estimator yields the lowest ISE for all sample sizes and thus has the lowest estimation error. In terms of PMSE, it is the best one for $n = 150$, whereas for $n = 500, 1000$ it performs comparably with SIGCOMP and PCA estimators. The performance of the AdaSStrue and AdaSS estimators is very similar in terms of ISE, whereas the AdaSStrue shows a lower PMSE. However, as expected, the effect of the knowledge of the true coefficient function tends to disappear as n increases, because the partial derivative estimates become more accurate.

3.2 Dampened harmonic motion function

This simulation scenario considers as coefficient function β the dampened harmonic motion function, also known as the *spring function* in the engineering literature. It is characterized by a sinusoidal behaviour with exponentially decreasing amplitude, that is

Table 1 The integrated squared error (ISE) and the prediction mean squared error (PMSE) for the TRU, SMOOTH, PCA, RIDGE, SIGCOMP, AdaSS and AdaSStrue estimators. The numbers outside the parentheses are the averages over 100 Monte Carlo replications, and the numbers inside parentheses are the corresponding standard errors. The values corresponding to the AdaSStrue estimator are emphasized in Italic the fact that they rely on the knowledge of the true coefficient function, which is unlikely in real applications. In bold are marked the lowest values among the AdaSS and the competitors

	$n = 150$		$n = 500$		$n = 1000$	
	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)
	Mexican hat					
TRU	0.4063 (0.0059)	0.3575 (0.0011)	0.1384 (0.0020)	0.3143 (0.0007)	0.0660 (0.0011)	0.3031 (0.0005)
SMOOTH	0.2191 (0.0020)	0.3382 (0.0007)	0.0917 (0.0008)	0.3088 (0.0005)	0.0564 (0.0006)	0.3027 (0.0005)
PCA	0.2519 (0.0068)	0.3234 (0.0007)	0.0681 (0.0013)	0.3030 (0.0005)	0.0368 (0.0008)	0.2995 (0.0005)
RIDGE	0.8813 (0.0083)	0.3629 (0.0008)	0.3542 (0.0041)	0.315 7 (0.0006)	0.1847 (0.0022)	0.3056 (0.0005)
SIGCOMP	0.1465 (0.0026)	0.3192 (0.0006)	0.053 2 (0.0006)	**0.3026** (0.0005)	0.0358 (0.0004)	0.2999 (0.0005)
AdaSS	**0.0856** (0.0023)	**0.3171** (0.0007)	**0.0359** (0.0010)	0.3027 (0.0005)	**0.0217** (0.0007)	**0.2994** (0.0005)
AdaSStrue	*0.0726 (0.0176)*	*0.3080 (0.0007)*	*0.0399 (0.0153)*	*0.2994 (0.0005)*	*0.0188 (0.0048)*	*0.2977 (0.0005)*
	Dampened harmonic					
TRU	0.2851 (0.0050)	0.5403 (0.0014)	0.0983 (0.0010)	0.5051 (0.0010)	0.0651 (0.0009)	0.4960 (0.0010)
SMOOTH	0.2288 (0.0042)	0.5391 (0.0013)	0.0836 (0.0007)	0.5032 (0.0010)	0.0555 (0.0005)	0.4936 (0.0010)
PCA	0.3710 (0.0093)	0.5259 (0.0012)	0.1100 (0.0020)	**0.4994** (0.0010)	0.0594 (0.0011)	**0.4915** (0.0010)
RIDGE	1.4221 (0.0135)	0.5925 (0.0016)	0.6082 (0.0076)	0.5203 (0.0011)	0.3271 (0.0038)	0.5014 (0.0010)
SIGCOMP	0.2541 (0.0045)	**0.5221** (0.0012)	0.1235 (0.0013)	0.5018 (0.0010)	0.0942 (0.0009)	0.4950 (0.0010)
AdaSS	**0.1749** (0.0038)	0.5241 (0.0012)	**0.0695** (0.0012)	0.4997 (0.0010)	**0.0461** (0.0008)	0.4918 (0.0010)
AdaSStrue	*0.1504 (0.0030)*	*0.5179 (0.0012)*	*0.0744 (0.0018)*	*0.4985 (0.0010)*	*0.0582 (0.0022)*	*0.4912 (0.0010)*
	Rapid change					

Table 1 continued

	$n = 150$		$n = 500$		$n = 1000$	
	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)	ISE ($\times 10^{-1}$)	PMSE ($\times 10^{-2}$)
TRU	1.9910 (0.0278)	4.0461 (0.0001)	0.9178 (0.0100)	3.7583 (0.0001)	0.6020 (0.0074)	3.6989 (0.0001)
SMOOTH	1.2961 (0.0133)	3.9427 (0.0001)	0.5738 (0.0046)	3.7205 (0.0001)	0.3590 (0.0027)	3.6787 (0.0001)
PCA	5.1052 (0.0971)	4.3070 (0.0001)	1.5870 (0.0271)	3.7978 (0.0001)	0.8383 (0.0125)	3.7141 (0.0001)
RIDGE	10.4781 (0.1059)	4.4295 (0.0001)	4.1991 (0.0537)	3.8459 (0.0001)	2.2250 (0.0278)	3.7356 (0.0001)
SIGCOMP	1.7129 (0.0209)	4.0352 (0.0001)	0.8615 (0.0234)	3.7702 (0.0001)	0.8552 (0.0167)	3.7428 (0.0001)
AdaSS	**1.0482** (0.0166)	**3.8737** (0.0001)	**0.4526** (0.0077)	**3.6928** (0.0001)	**0.2916** (0.0044)	**3.6662** (0.0001)
AdaSStrue	*0.8181 (0.0191)*	*3.8274 (0.0001)*	*0.3434 (0.0080)*	*3.6759 (0.0001)*	*0.2114 (0.0050)*	*3.6541 (0.0001)*

The bold indicates the best method excluding AdaSStrue (in Italic) that is a benchmark method that could be not used in practice

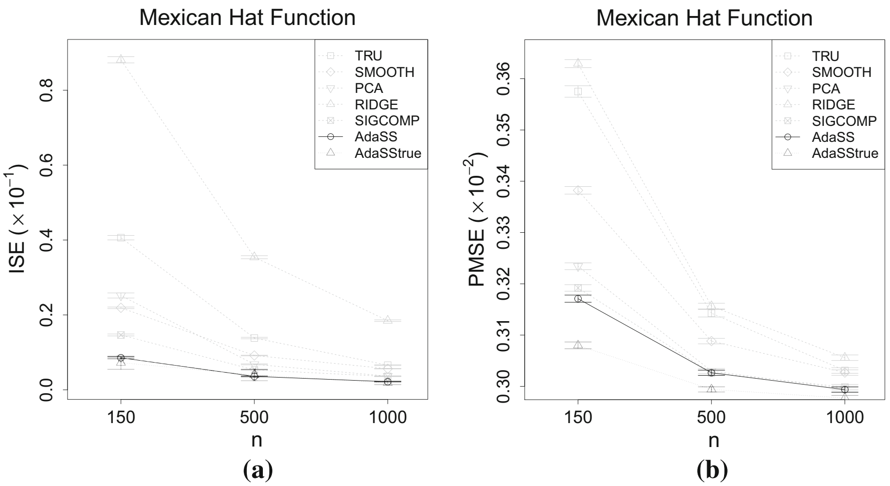

Fig. 2 **a** The integrated squared error (ISE) and **b** The prediction mean squared error (PMSE) $\pm standard\ error$ for the TRU, SMOOTH, PCA, RIDGE, SIGCOMP, AdaSS and AdaSStrue estimators in the case of the Mexican hat function

$$\beta(s,t) = 1 + 5\exp\left[-5(s+t)\right]\left[\cos(10\pi s) + \cos(10\pi t)\right], \quad s,t \in [0,1] \times [0,1].$$

Figure 3 displays the AdaSS and the SMOOTH estimates along with the true coefficient function. Also in this scenario, the AdaSS estimates are smoother than the SMOOTH estimates in regions of small curvature. But, it is more flexible where the coefficient function is more wiggly. Note that intuitively, the SMOOTH estimator trades off its smoothness over the whole domain. Indeed, it over-smooths at small values of s and t and under-smooths elsewhere.

In Fig. 4 and in the second tier of Table 1, values of the ISE and PMSE for the AdaSS, AdaSStrue, and competitor estimators are shown as a function of the sample size n. Even in this case, the AdaSS estimator achieves the lowest ISE for all sample sizes, and thus, the lowest estimation error, without taking into account the AdaSStrue estimator. Strictly speaking, in terms of PMSE, note that the proposed estimator is not always the best choice, but it shows only a small difference with the best methods, viz., PCA and SIGCOMP estimators. In this case, the AdaSS and AdaSStrue performance is very similar for $n = 500, 1000$, whereas, for $n = 150$, the AdaSStrue performs slightly better especially in terms of PMSE.

3.3 Rapid change function

In this scenario, the true coefficient function β is obtained by the rapid change function, that is

$$\beta(s,t) = 1 - \frac{5}{1 + \exp\left[10(s+t-0.2)\right]} + \frac{5}{1 + \exp\left[75(s+t-0.8)\right]},$$
$$s,t \in [0,1] \times [0,1].$$

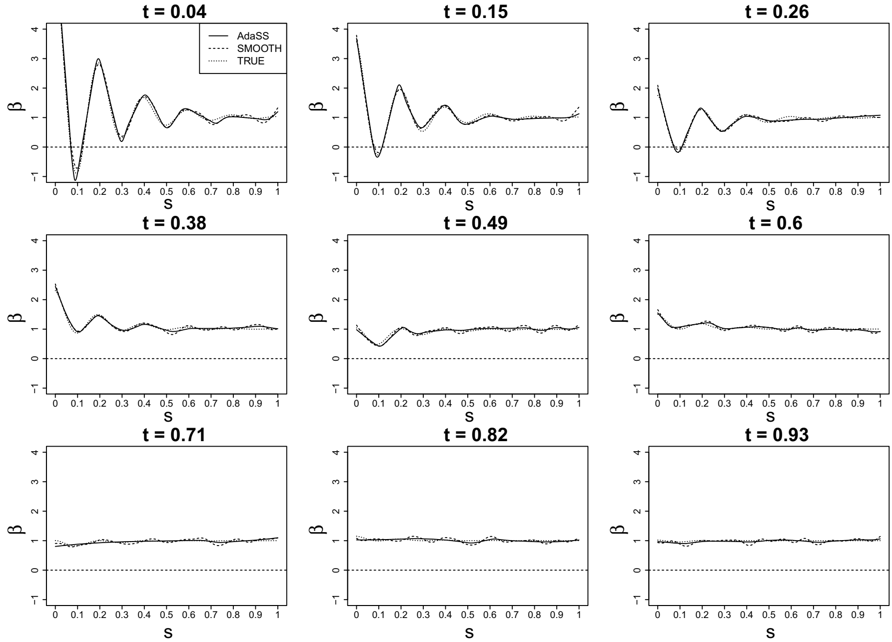

Fig. 3 AdaSS (solid line) and SMOOTH (dashed line) estimates of the coefficient functions and the TRUE coefficient function β (dotted line) for different values of t in the case of the dampened harmonic motion function

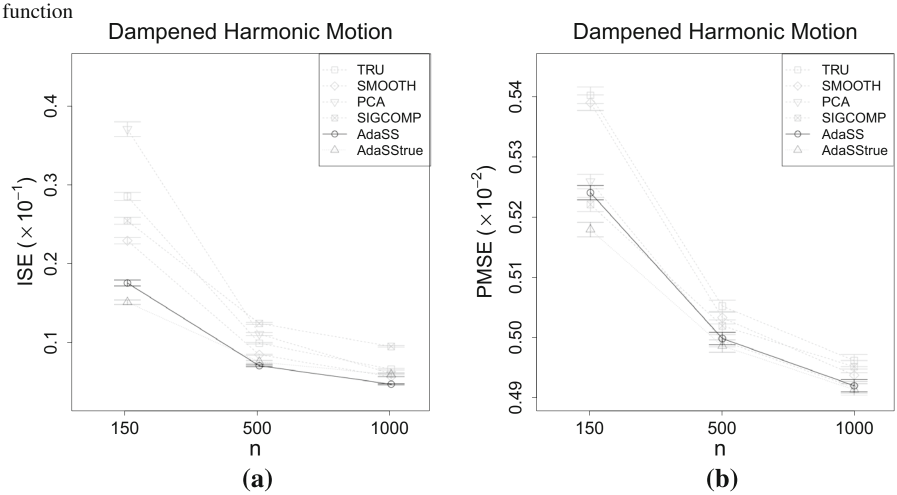

Fig. 4 **a** The integrated squared error (ISE) and **b** The prediction mean squared error (PMSE) $\pm standard\ error$ for the TRU, SMOOTH, PCA, SIGCOMP, AdaSS and AdaSStrue estimators in the case of the dampened harmonic motion function. The Ridge estimator is not considered due to its too different performance

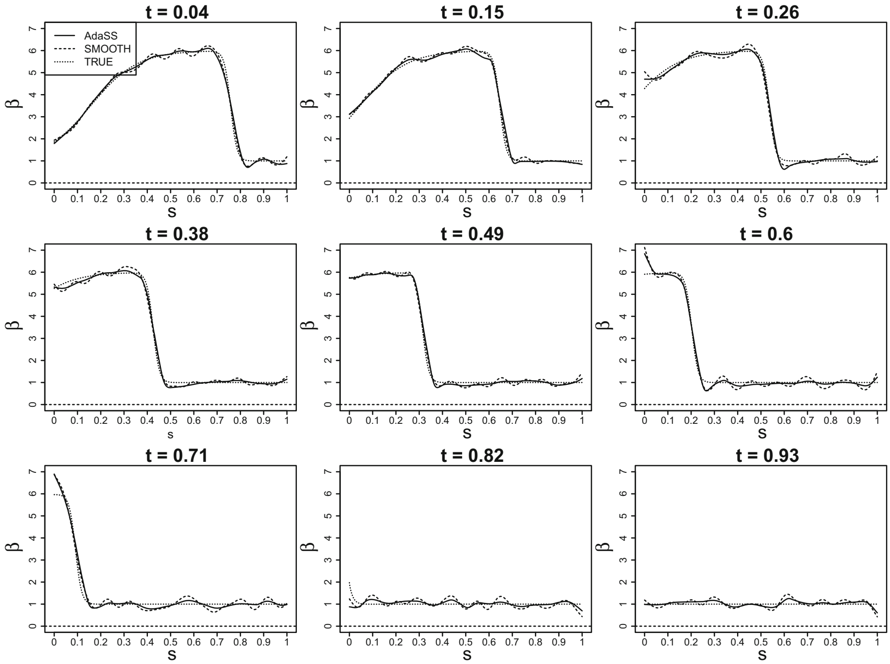

Fig. 5 AdaSS (solid line) and SMOOTH (dashed line) estimates of the coefficient functions and the TRUE coefficient function β (dotted line) for different values of t in the case of the rapid change function

Figure 5 shows the AdaSS and SMOOTH estimate when β is the rapid change function. The SMOOTH estimate is rougher than the AdaSS one in regions that are far from the rapid change point. On the contrary, the AdaSS estimate is able to be smoother in the flat region and to be as accurate as the SMOOTH estimate near the rapid change point.

In Fig. 6 and the third tier of Table 1, values of the ISE and PMSE for the AdaSS, AdaSStrue, and competitor estimators are shown for sample sizes $n = 150, 500, 1000$. The AdaSS estimator outperforms the competitors, both in terms of ISE and PMSE. The performance of the AdaSStrue estimator is slightly better than that of the AdaSS one and this difference in performance reduces as n increases.

4 Real-data examples

In this section, two real datasets, namely *Swedish mortality* and *ship* CO_2 *emission* datasets, are considered to assess the performance of the AdaSS estimator in real applications.

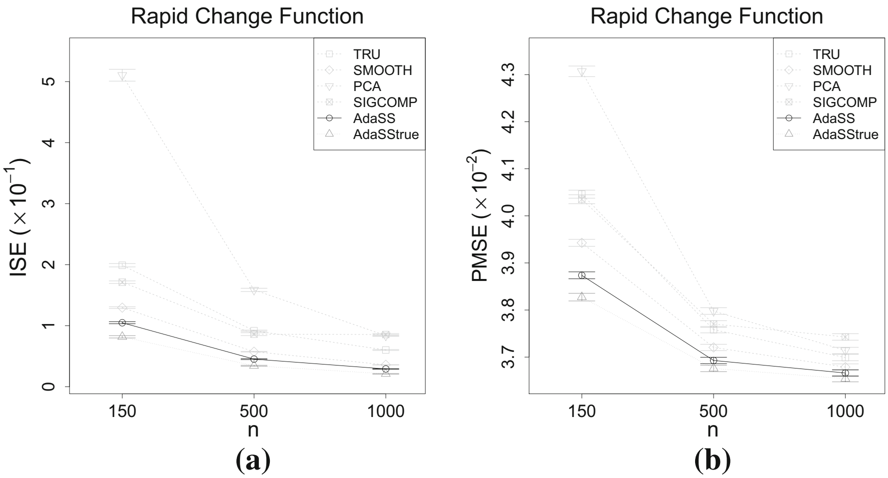

Fig. 6 **a** The integrated squared error (ISE) and **b** The prediction mean squared error (PMSE) $\pm standard\ error$ for the TRU, SMOOTH, PCA, SIGCOMP, AdaSS and AdaSStrue estimators in the case of the rapid change function. The Ridge estimator is not considered due to its too different performance

4.1 Swedish mortality dataset

The Swedish mortality dataset (available from the Human Mortality Database —http://mortality.org—) is very well known in the functional literature as a benchmark dataset. It has been analysed by Chiou and Müller (2009) and Ramsay et al. (2009), among others. In this analysis, we consider the log-hazard rate functions of the Swedish female mortality data for year-of-birth cohorts that refer to females born in the years 1751-1935 with ages 0-80. The value of a log-hazard rate function at a given age is the natural logarithm of the ratio of females who died at that age and the number of females alive of the same age. The 184 considered log-hazard rate functions (Chiou and Müller 2009) are shown in Fig. 7. Without loss of generality, they have been normalized to the domain [0, 1].

The functions from 1751 (1752) to 1934 (1935) are considered as observations X_i (Y_i) of the predictor (response) in (1), $i = 1, \ldots, 184$. In this way, the relationship between two consecutive log-hazard rate functions becomes the focus of the analysis. To assess the predictive performance of the methods considered in the simulation study (Sect. 3), for 100 times, 166 observations out of 184 are randomly chosen, as training set, to fit the model. The 18 remaining ones are used as a test set to calculate the PMSE. The averages and standard deviations of PMSEs are shown in the first line of Table 2. The AdaSS estimator outperforms all the competitors. Only the RIDGE estimator has comparable predictive performance.

Figure 8 shows the AdaSS estimates along with the RIDGE estimates that represents the best competitor methods in terms of PMSE. The proposed estimator has slightly better performance than the competitor, but, at the same time, it is much more interpretable. In fact, it is much smoother where the coefficient function seems to

Table 2 The prediction mean squared error (PMSE) for the TRU, SMOOTH, PCA, RIDGE, SIGCOMP, and AdaSS estimators. The numbers outside the parentheses are the averages of the PMSE over 100 replications, and the numbers inside parentheses are the corresponding standard errors

	TRU	SMOOTH	PCA	RIDGE	SIGCOMP	AdaSS
Swedish mortality ($\times 10^{-2}$)	0.7373 (0.0000)	0.5938 (0.0000)	0.6131 (0.0000)	0.5749 (0.0000)	1.0173 (0.0000)	**0.5706** (0.0000)
Ship CO^2 emission	0.1019 (0.0008)	0.0814 (0.0007)	0.0689 (0.0008)	**0.0625** (0.0007)	0.1033 (0.0013)	0.0771 (0.0007)

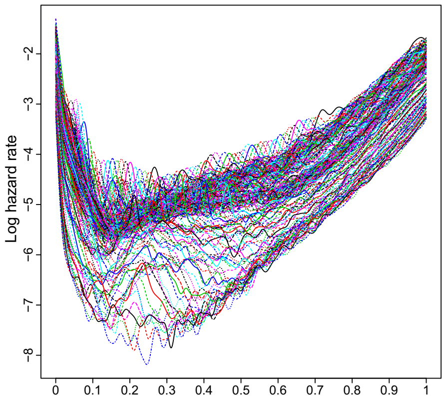

Fig. 7 Log-hazard rate functions for Swedish female cohorts from 1751 to 1935

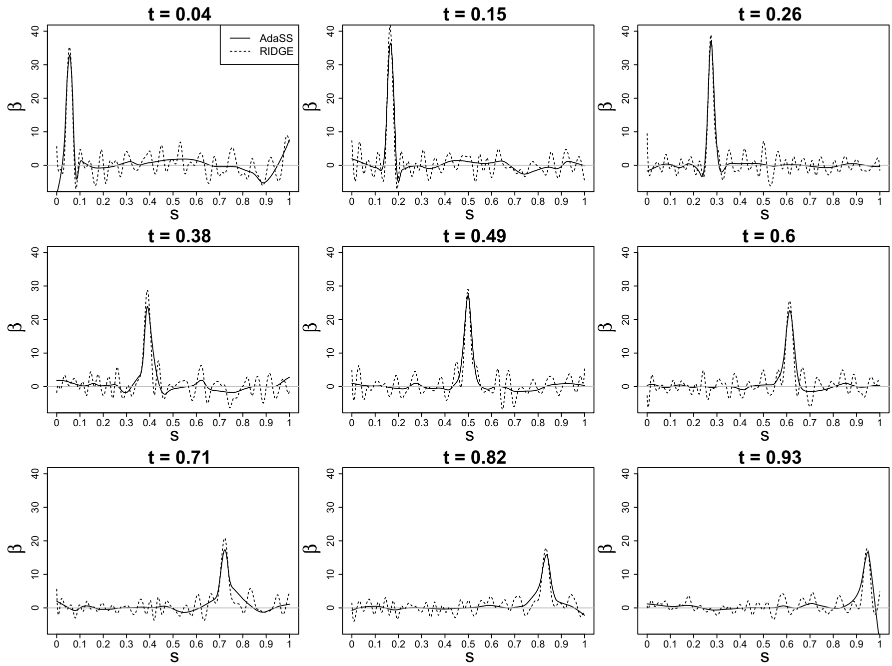

Fig. 8 AdaSS (solid line) and RIDGE (dashed line) estimates of the coefficient functions for different values of t in the Swedish Mortality dataset

be mostly flat and successfully captures the pattern of β in the peak region. On the contrary, the RIDGE estimates are particularly rough over regions of low curvature. The optimal tuning parameters selected for the AdaSS estimates depicted in Fig. 8 are $\lambda_s^{AdaSS} = 10^{2.16}$, $\lambda_t^{AdaSS} = 10^0$, $\tilde{\delta}_s = 0.003$, $\tilde{\delta}_t = 0.052$, $\gamma_s = 2.46$, $\gamma_t = 3.60$, where $\tilde{\delta}_s = \delta_s / \max |\widehat{\beta_s^{m_s}}(s,t)|$ and $\tilde{\delta}_t = \delta_t / \max |\widehat{\beta_t^{m_t}}(s,t)|$.

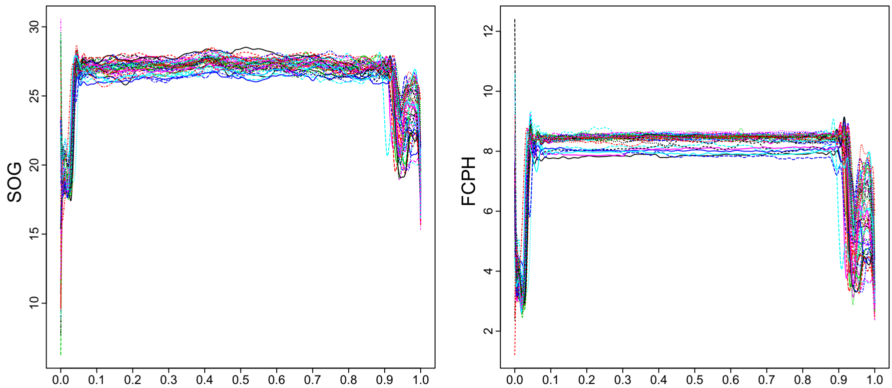

Fig. 9 SOG and FCPH observations from a Ro-Pax ship

4.2 Ship CO_2 emission dataset

The ship CO_2 emission dataset has been thoroughly studied in the very last years (Lepore et al. 2018; Reis et al. 2020; Capezza et al. 2020; Centofanti et al. 2021). It was provided by the shipping company Grimaldi Group to address aspects related to the issue of monitoring fuel consumptions or CO_2 emissions for a Ro-Pax ship that sails along a route in the Mediterranean Sea. In particular, we focus on the study of the relation between the *fuel consumption per hour* (FCPH), assumed as the response, and the *speed over ground* (SOG), assumed as the predictor. The observations considered were recorded from 2015 to 2017. Figure 9 shows the 44 available observations of SOG and FCPH (Centofanti et al. 2021).

Similarly to the Swedish mortality dataset, the prediction performance of the methods are assessed by randomly chosen 40 out of 44 observations to fit the model and by using the 4 remaining observations to compute the PMSE. This is repeated 100 times. The averages and standard deviations of the PMSEs are listed in the second line of Table 2. The AdaSS estimator is, in this case, outperformed by the RIDGE estimator, which achieves the lowest PMSE. However, as shown in Fig. 10, it is able both to well estimate the coefficient function over peaky regions, as the RIDGE estimator, and to smoothly adapt over the remaining part of the domain. Also, the PCA estimator achieves a smaller PMSE than that of the proposed estimator. However, the PCA estimator is even rougher than the RIDGE estimator and, thus, it is not shown in Fig. 10. The optimal tuning parameters selected for the AdaSS estimates depicted in Fig. 10 are $\lambda_s^{AdaSS} = 10^{1.93}$, $\lambda_t^{AdaSS} = 10^{0.49}$, $\tilde{\delta}_s = 0.06$, $\tilde{\delta}_t = 0.01$, $\gamma_s = 2.22$, $\gamma_t = 2.22$.

5 Conclusion

In this article, the AdaSS estimator is proposed for the function-on-function linear regression model where each value of the response, at any domain point, depends linearly on the full trajectory of the predictor. The introduction of two adaptive smoothing penalties, based on an initial estimate of its partial derivatives, allows the proposed

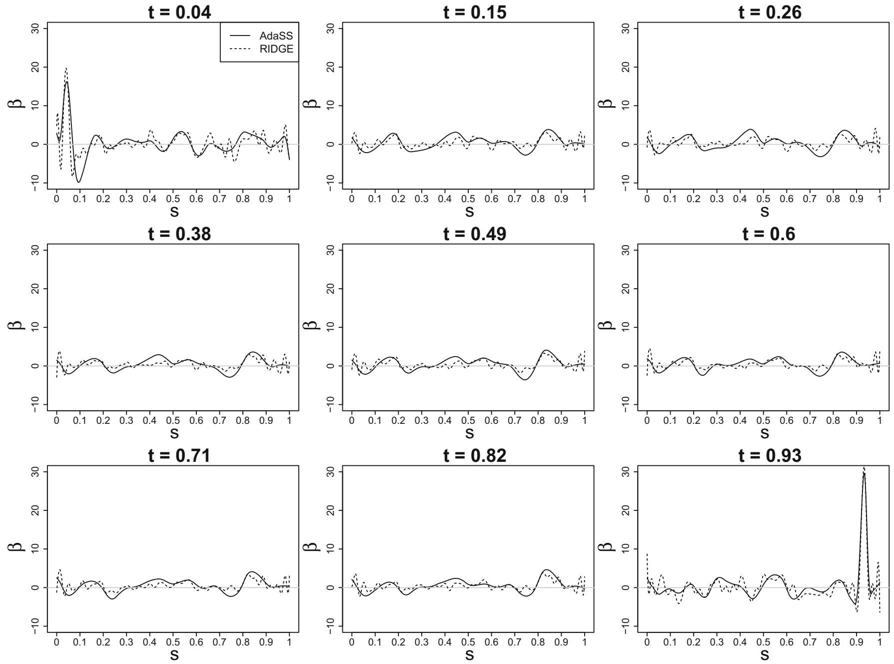

Fig. 10 AdaSS (solid line) and RIDGE (dashed line) estimates of the coefficient functions for different values of t in the ship CO_2 emission dataset

estimator to better adapt to the coefficient function. By means of a simulation study, the proposed estimator has proven favourable performance with respect to those achieved by the five competitors already appeared in the literature before, both in terms of estimation and prediction error. The adaptive feature of the AdaSS estimator is advantageous for the interpretability of the results with respect to the competitors. Moreover, its performance has shown to be competitive also with respect to the case where the true coefficient function is known. Finally, the proposed estimator has been successfully applied to real-data examples considered, viz., the Swedish mortality and ship CO_2 emission datasets. However, some challenges are still open. Even though the proposed evolutionary algorithm has shown to perform particularly well both in the simulation study and the real-data examples, the choice of the tuning parameters still remains in fact a critical issue, because of the curse of dimensionality. This could be even more problematic in the perspective of extending the AdaSS estimator to the FoF regression model with multiple predictors.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1007/s00180-022-01223-6. Supplementary Material is available online and contains the derivation of the approximation of the AdaSS penalty terms (Supplementary Material 1), and additional simulation studies to assess the performance of both the proposed estimator for different choices of partial derivative estimates (Supplementary Material 2.1) and the repeated application of the AdaSS estimation method (Supplementary Material 2.2). Supplementary Material 3 presents the optimal tuning parameter selected in the simulation study in Section 3.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00180-022-01223-6.

Acknowledgements The authors are deeply grateful to the editor, the associate editor, and two referees for their reviews, which led to significant improvements of the manuscript.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

Declarations

Data Availability Statement The code used in the simulation study is included in the supplementary information. The Swedish mortality dataset is available from the Human Mortality Database (http://mortality.org). For confidentiality reasons, the ship CO_2 emission dataset is available upon request.

Conflict of interest The authors declare that they have no conflict of interest.

References

Abramovich F, Steinberg DM (1996) Improved inference in nonparametric regression using lk-smoothing splines. J Stat Plann Inference 49(3):327–341

Bäck T, Fogel DB, Michalewicz Z (1997) Handbook of evolutionary computation. Chapman and Hall/CRC, UK

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(10):281–305

Bergstra J, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst, Curran Assoc Inc 24:871

Canale A, Vantini S (2016) Constrained functional time series: applications to the italian gas market. Int J Forecast 32(4):1340–1351

Capezza C, Lepore A, Menafoglio A, Palumbo B, Vantini S (2020) Control charts for monitoring ship operating conditions and CO_2 emissions based on scalar-on-function regression. Appl Stoch Models Business Ind 36(3):477–500

Capezza C, Centofanti F, Lepore A, Menafoglio A, Palumbo B, Vantini S (2022) Functional regression control chart for monitoring ship CO_2 emissions. Quality Reliability Eng Int 38(3):1519–1537

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

Centofanti F, Fontana M, Lepore A, Vantini S (2020) Smooth lasso estimator for the function-on-function linear regression model. http://arxiv.org/abs/200700529

Centofanti F, Lepore A, Menafoglio A, Palumbo B, Vantini S (2021) Functional regression control chart. Technometrics 63(3):281–294

Chiou JM, Müller HG (2009) Modeling hazard rates as functional data for the analysis of cohort lifetables and mortality forecasting. J Am Stat Assoc 104(486):572–585

Eubank RL (1999) Nonparametric regression and spline smoothing. Chapman and Hall/CRC, UK

Ferraty F, Vieu P (2006) Nonparametric functional data analysis: theory and practice. Springer, New York
Green PJ, Silverman BW (1993) Nonparametric regression and generalized linear models: a roughness penalty approach. Chapman and Hall/CRC, UK
Gu C (2013) Smoothing spline ANOVA models. Springer, New York
Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction. Springer, New York
Herlihy M, Shavit N (2011) The art of multiprocessor programming. Morgan Kaufmann, USA
Horváth L, Kokoszka P (2012) Inference for functional data with applications. Springer, New York
Hsing T, Eubank R (2015) Theoretical foundations of functional data analysis, with an introduction to linear operators. Wiley, New Jersey
Hullait H, Leslie DS, Pavlidis NG, King S (2021) Robust function-on-function regression. Technometrics 63(3):396–409
Ivanescu AE, Staicu AM, Scheipl F, Greven S (2015) Penalized function-on-function regression. Comput Stat 30(2):539–568
Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K (2017) Population based training of neural networks. http://arxiv.org/abs/171109846
James GM (2002) Generalized linear models with functional predictors. J R Stat Soc: Ser B (Stat Methodol) 64(3):411–432
James GM, Wang J, Zhu J et al (2009) Functional linear regression that's interpretable. Annal Stat 37(5A):2083–2108
Kokoszka P, Reimherr M (2017) Introduction to functional data analysis. Chapman and Hall/CRC, UK
Lepore A, Palumbo B, Capezza C (2018) Analysis of profiles for monitoring of modern ship performance via partial least squares methods. Quality Eng Int 34(7):1424–1436
Lin Z, Cao J, Wang L, Wang H (2017) Locally sparse estimator for functional linear regression models. J Comput Gr Stat 26(2):306–318
Luo R, Qi X (2017) Function-on-function linear regression by signal compression. J Am Stat Assoc 112(518):690–705
Luo R, Qi X (2019) Interaction model and model selection for function-on-function regression. J Comput Gr Stat 28(2):1–14
Morris JS (2015) Functional regression. Ann Rev Stat Appl 2:321–359
Müller HG, Stadtmüller U (2005) Generalized functional linear models. Annal. Stat 33(2):774–805
Palumbo B, Centofanti F, Del Re F (2020) Function-on-function regression for assessing production quality in industrial manufacturing. Quality Reliability Eng Int 36(8):2738–2753
Pintore A, Speckman P, Holmes CC (2006) Spatially adaptive smoothing splines. Biometrika 93(1):113–125
Prchal L, Sarda P (2007) Spline estimator for the functional linear regression with functional response. Preprint
Qi X, Luo R (2018) Function-on-function regression with thousands of predictive curves. J Multiv Anal 163:51–66
Qi X, Luo R (2019) Nonlinear function on function additive model with multiple predictor curves. Stat Sinica 29:719–739
Ramsay J, Silverman B (2005) Functional data analysis. Springer, New York
Ramsay JO, Hooker G, Graves S (2009) Functional data analysis with R and MATLAB. Springer, New York
Reis MS, Rendall R, Palumbo B, Lepore A, Capezza C (2020) Predicting ships' CO_2 emissions using feature-oriented methods. Appl Stoch Models Business Ind 36(1):110–123
Ruppert D, Carroll RJ (2000) Theory & methods: spatially-adaptive penalties for spline fitting. Australian & New Zealand J Stat 42(2):205–223
Scheipl F, Greven S (2016) Identifiability in penalized function-on-function regression models. Electron J Stat 10(1):495–526
Scheipl F, Staicu AM, Greven S (2015) Functional additive mixed models. J Comput Gr Stat 24(2):477–501
Schumaker L (2007) Spline functions: basic theory. Cambridge University Press, Cambridge
Storlie CB, Bondell HD, Reich BJ (2010) A locally adaptive penalty for estimation of functions with varying roughness. J Comput Gr Stat 19(3):569–589
Wahba G (1990) Spline models for observational data. Soc Ind Appl Math 2:61
Wang X, Du P, Shen J (2013) Smoothing splines with varying smoothing parameter. Biometrika 100(4):955–970
Yang L, Hong Y (2017) Adaptive penalized splines for data smoothing. Comput Stat Anal 108:70–83

Yao F, Müller HG (2010) Functional quadratic regression. Biometrika 97(1):49–64
Yao F, Müller HG, Wang JL (2005) Functional data analysis for sparse longitudinal data. J Am Stat Assoc 100(470):577–590
Yao F, Müller HG, Wang JL (2005) Functional linear regression analysis for longitudinal data. Annal Stat 33(6):2873–2903
Zhou J, Wang NY, Wang N (2013) Functional linear model with zero-value coefficient function at subregions. Stat Sinica 23(1):25