Computers and Operations Research 123 (2020) 105050

Contents lists available at ScienceDirect

Computers and Operations Research

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

Selection of time instants and intervals with Support Vector Regression for multivariate functional data

Rafael Blanquero [a], Emilio Carrizosa [a], Asunción Jiménez-Cordero [b,*], Belén Martín-Barragán [c]

[a] Departamento de Estadística e Investigación Operativa and Instituto de Matemáticas de la Universidad de Sevilla (IMUS). Facultad de Matemáticas. C/ Tarfia s/n 41012, University of Seville, Seville, Spain
[b] Group OASYS. Ada Byron Research Building, C/ Arquitecto Francisco Peñalosa, 18, 29010, University of Málaga, Málaga, Spain
[c] Business School, 29 Buccleuch Place, EH89JS, University of Edinburgh, Edinburgh, UK

ARTICLE INFO

Article history:
Received 10 September 2018
Revised 13 March 2020
Accepted 3 July 2020
Available online 19 July 2020

Keywords:
Machine learning
Functional regression
Support Vector Regression
Time interval selection

ABSTRACT

When continuously monitoring processes over time, data is collected along a whole period, from which only certain time instants and certain time intervals may play a crucial role in the data analysis. We develop a method that addresses the problem of selecting a finite and small set of short intervals (or instants) able to capture the information needed to predict a response variable from multivariate functional data using Support Vector Regression (SVR).

In addition to improving interpretability, storage requirements, and monitoring cost, feature selection can potentially reduce overfitting by mitigating data autocorrelation. We propose a continuous optimization algorithm to fit the SVR parameters and select intervals and instants. Our approach takes advantage of the functional nature of the data by formulating a new bilevel optimization problem that integrates selection of intervals and instants, tuning of some key SVR parameters and fitting the SVR. We illustrate the usefulness of our proposal in some benchmark data sets.

1. Introduction

Functional data analysis (FDA) (Ferraty and Vieu, 2006; Ramsay and Silverman, 2002, 2005), is an extension of the classic multivariate analysis which is particularly oriented to handle observations of functional nature, where each observation represents the dependency of a varying quantity on another quantity whose values vary over a given interval. When each observation consists of more than one function, we call it multivariate functional data analysis. The applications of FDA span a wide range of fields such as chemical processes, Blanquero et al. (2016a,b), meteorology, Martín-Barragán et al. (2014), speech recognition, Rossi and Villa (2006) and spectrometry, Ferraty et al. (2010), Hernández et al. (2007) and Martín-Barragán et al. (2014), among others.

In this paper, we focus on functional regression, Ferraty et al. (2010), Hernández et al. (2007), James et al. (2009), Kneip et al. (2016) and Müller and Stadtmüller (2005), one of the most challenging problems in FDA. Even though the majority of the literature on functional regression is focused on the univariate counterpart, in this paper we are interested in the prediction of a scalar response, based on the information provided by multivariate functional data.

Predictor-response relationships are harder to be found and interpret as the dimension of the data becomes larger, or even infinite, as in FDA. Selecting, without damaging the predictive ability, a few short subintervals from the full monitoring interval would definitely lead to a much better understanding of the data, enhancing quicker predictions and easing decision making. Note that time instants can be considered as a degenerate case of intervals whose length is zero. Hence, unless explicitly stated, we will use in this text the term interval as a general term encompassing both time intervals and time instants. The replacement of the whole interval with a few short subintervals or even degenerate can be seen as a variable selection procedure within an infinite-dimensional framework.

In the literature of feature selection in finite dimensional data, we can find a wide variety of works. See Chandrashekar and Sahin (2014), Guyon and Elisseeff (2003) and Molina et al. (2002) for a survey. Particularly, in the regression area, we can highlight (Andersen and Bro, 2010; Karagiannopoulos et al., 2007; Li et al., 2009; Mehmood et al., 2012; Van Dijck and Van Hulle, 2006; Yang and Ong, 2011; Zhang, 2009). Variable selection has also been studied in other domains such as clustering (Dash et al., 2002;

* Corresponding author.
E-mail addresses: rblanquero@us.es (R. Blanquero), ecarrizosa@us.es (E. Carrizosa), asuncionjc@uma.es (A. Jiménez-Cordero), belen.martin@ed.ac.uk (B. Martín-Barragán).

https://doi.org/10.1016/j.cor.2020.105050
0305-0548/

Dash and Liu, 2000; Dash et al., 1997; Law et al., 2004, 2003; Li et al., 2008) and supervised classification (Aytug, 2015; Benítez-Peña et al., 2019; Bertolazzi et al., 2016; Carrizosa et al., 2011; Dash and Liu, 1997; Maldonado and Weber, 2009; Maldonado et al., 2011).

Feature selection methods are grouped into one of the following categories (Torrecilla Noguerales, 2015):

- In *filter methods* feature selection is done as a preprocessing step regardless of the model or method that will be used to predict. See, for instance, Lazar et al. (2012), for a review of these approaches. Most of these methods rank the variables according to a relevance measure and discard the least relevant features. Advanced versions of filter methods include the use of sparse PCA, Johnstone and Lu (2009), and SIR (Slice Inverse Regression), Picheny et al. (2019). Both PCA and SIR produce transformations of the feature space that are typically used for dimensionality reduction, but these transformations usually involve the whole set of original features. Their sparse versions aim to achieve the same reduction but using fewer features to allow for interpretability. The final transformed features can then be plugged into any regression model to obtain the final predictor.
- *Wrapper methods* Kohavi and John (1997) interact with the predictive model or method, but use them as a black box to assign scores to the features. From a computational point of view, they are less efficient than filter methods, although, as a counterpart, may provide better accuracy.
- *Embedded methods* take the interaction a step forward, and integrate the estimation of the prediction model and the feature selection steps. Hence, model construction and feature selection are done jointly, usually through the resolution of a single optimization problem. See Maldonado and Weber (2010) for an example of an embedded method with SVR as a regression algorithm.
- *Hybrid methods* combine filter and wrapper approaches, usually in two-steps algorithms (Hsu et al., 2011; Hua et al., 2009).

Several traditional multivariate statistical approaches to address regression problems have been extended to FDA. Since the early days of FDA, least squares methods have been applied to linear regression with functional predictors, Ramsay and Dalzell (1991) and Ramsay and Silverman (2005). Alternative techniques, such as Support Vector Regression (SVR) (Smola and Schölkopf, 2004), have also been extended to cope with functional data in a nonlinear manner (Hernández et al., 2007, 2009). Some dimensionality reduction techniques for linear regression in FDA have been introduced, such as principal component regression or partial least squares, Aguilera et al. (1997), Delaigle and Hall (2012), Preda and Saporta (2005,) and Reiss and Ogden (2007). These are projection-based methods that reduce dimensionality where the retained features are a combination of (potentially) all the original variables in the model. In this sense, they select neither time instants nor short intervals. An increasing number of references have recently tackled dimensionality reduction via variable selection. Most of them have considered the selection of time instants. For example, the algorithm proposed in James et al. (2009) focused on the interpretability by selecting time instants. Since such selection is based on the ℓ_1-norm regularization, it is difficult to control the number of selected instants. In Kneip et al. (2016) a method is proposed to detect the most important points of impact among a predefined set of time instants, in which the functional data are measured, i.e., it is assumed that the impact points belong only to the set of timestamps where the functions are monitored, which is not always the case. Moreover, Kneip et al. (2016) is a generalization of the model proposed in McKeague and Sen (2010) where the identifiability and estimation of just one time instant is sought. The work of Aneiros and Vieu (2014) directly applies standard multivariate procedures to discretized functional data. Thus, the functional nature of the data is disregarded and not exploited.

Functional nonparametric regression models have also been proposed, Ferraty et al. (2010), Ferraty and Vieu (2004, 2006) and Rossi and Conan-Guez (2005), and the optimal selection of the time instants has been studied too. We can highlight, for instance, the works of Aneiros and Vieu (2016) and Ferraty et al. (2010), where the most influential design points are sought among a given (large) set, usually hard to obtain, or the methodologies of Berrendero et al. (2019) and Ferraty et al. (2010) based on a greedy approach, in which the time instants are sequentially located.

Nevertheless, in many real-life applications, the important information may not be (only) represented by isolated time instants, but also by some influential time intervals. Hence, when the predictive ability of the models that use only isolated time instants is not acceptable, the selection of a few short intervals may provide an alternative that is still easy to interpret. Note that there is an important conceptual difference between time instants selection and time interval selection in functional data. In the former, the infinite dimensional space representing the functional data is reduced to a finite dimensional space. Hence, multivariate techniques can be then applied. In contrast, when selecting time intervals, the reduced space will still be infinite-dimensional. In such a case, the estimation of the final model will need FDA techniques. This difficulty does not hurt interpretability if only a few intervals are selected and their lengths are small. Very few methods have been proposed in the literature to deal with the selection of time intervals in functional data regression. The authors of Tutz and Gertheiss (2010), Grollemund et al. (2019) and Park et al. (2016) focus on the linear regression case. The article Tutz and Gertheiss (2010) proposes a linear model for selecting a group of variables with a predefined length. This strategy yields intervals of the same size, which makes the model less flexible. The strategy proposed in Grollemund et al. (2019) pursues time intervals which may overlap, under the assumption of a linear regression model and normal distribution probability. In addition, the work in Park et al. (2016) also assumes a linear regression model and proposes an aggregative framework; it starts from a given initial partition of the domain and the intervals are then joined two at a time according to their prediction ability. Apart from the difficulties to obtain a good initial partition, the number of possible combinations of two intervals may explode if a large number of initial intervals are available. There are other two proposals that, without aiming directly with the selection of intervals, can be adapted for that purpose. The more general problem of selecting functional features (functions) from a given set is addressed in Fraiman et al. (2016) for linear regression with scalar or functional response, functional supervised and unsupervised classification, and functional PCA. If the set of functions is chosen as the *local averages*, the method will select intervals. However, that method is very inflexible as the intervals need to be predefined, and the function considered will be constant at such an interval. Another method that can select intervals, for an adequate choice of the regularization, is proposed in James et al. (2009). However, neither the number of intervals nor their length is directly penalized. Hence they are difficult to control. Moreover, this methodology only considers the linear regression model. Finally, the nonlinear nonparametric case is addressed in Picheny et al. (2019), where a sparse version of SIR for functional data is proposed. As mentioned before for SIR, the proposal in Picheny et al. (2019) is a filtering approach that selects the intervals without using the information about the nonlinear model. Indeed, their proposal seeks select intervals with the aim of replicating the results of SIR. As far as we are aware of,

this is the only proposal that has considered a nonlinear regression model. In particular, the feature selection problem in SVR with functional data has not yet been studied in the literature. For the classification problem, an SVM-based method to select time instants has been proposed in Blanquero et al. (2019b).

In this article, we consider the problem of the optimal selection of time intervals (including time instants as a degenerate case) for functional Support Vector Regression. We extend the work done recently for selecting time instants, but not intervals, for the classification, instead of regression, problem (Blanquero et al., 2019b). The use of Support Vector Regression allows us to capture nonlinearities. This is the first embedded feature selection method for this problem that is able to capture nonlinearities. The only other nonlinear alternative is a filtering method (Picheny et al., 2019). Furthermore, following the scheme of Blanquero et al. (2019b), our proposal handles in the very same way univariate and multivariate functions. Indeed, this is the first time the feature selection problem is considered in the context of multivariate functional regression. We formulate an optimization problem able to find at once the most important intervals of the multivariate functional data in terms of prediction. This issue is tackled via a penalization in the objective function which regulates the length of the intervals. Taking advantage of the functional behavior of the data, the selected time intervals are represented in the optimization model as continuous decision variables. Therefore, the so-obtained optimization problem may be solved by means of continuous optimization techniques. If instead, the data had been treated as multivariate finite-dimensional data, combinatorial optimization problems, which are very hard to solve due to the exponential number of candidate solutions, would have been obtained.

It is important to remark that the methodology of Blanquero et al. (2019b) is restricted to the search of time instants. In contrast, in this paper, we generalize such a model to seek time intervals and time instants (intervals of length zero).

The remainder of this paper is structured as follows. Section 2 is devoted to some previous definitions related to the use of the derivatives in the SVR problem. In Section 3 we detail the problem formulation and the solution approach, as well as the way to choose the best number of time intervals (including instants as a degenerate case). Section 4 describes the numerical experiments and Section 5 presents the results obtained with our approach. We finish in Section 6 with some conclusions.

2. Preliminaries

Here, we introduce the main definitions and concepts concerning our proposal. Section 2.1 details the notation of the multivariate functional data, including how to infer the higher-order information by means of the derivatives. Section 2.2 outlines the Support Vector Regression problem, and presents the kernel function here utilized.

2.1. Notation and derivatives management

Let s be a sample of individuals $\{(X_i, Y_i)\}_{i\in s}$, where $X_i \in \mathcal{X} = \mathcal{F}^p$ is formed by p functional components, $X_i(t) = \left(X_{i1}(t), \ldots, X_{ip}(t)\right)$ with $X_{iv} : [0,T] \to \mathbb{R}$ belonging to the class $\mathcal{F}$ of d-times continuously differentiable functions on the time interval $[0,T]$, and $Y_i \in \mathbb{R}$. The main goal is to find a rule able to predict the response $Y \in \mathbb{R}$ from the information of the multivariate functional data $X \in \mathcal{X}$.

Our proposal is applicable to pure multivariate functional data, as well as to univariate functional data, $X(t) \in \mathcal{F}$. The simplest way would be just to consider $p = 1$. However, a more sophisticated approach can be achieved if the derivatives are applied to transform the univariate in multivariate data. Particularly, a univariate datum $X(t)$ is transformed in:

$$\left(X(t), X'(t), \ldots, X^{d)}(t)\right) \tag{1}$$

with $X^{d)}(t)$ denoting the derivative of degree d of $X(t)$. In this way, the higher-order information provided by the derivatives can also be included in the pure multivariate functional data, $X(t) \in \mathcal{F}^p$, as follows:

$$\left(X_1(t), \ldots, X_p(t), X_1'(t), \ldots, X_p'(t), \ldots, X_1^{d)}(t), \ldots, X_p^{d)}(t)\right) \tag{2}$$

The usage of the derivatives may be decisive to obtain accurate predictions, as is shown in Section 4.

In real-life applications, the original functional data X_i are only known in some time instants. Hence, smoothing techniques, e.g. De Boor (1978) and Friedman et al. (2001), should be applied as a preprocessing step so that an approximation to the original function X_i can be obtained from the observed time instants.

Moreover, if higher-order information is taken into account, one can first compute the finite increments, and then smooth the sequence of increments. As an example, the first derivative of $X(t)$ at a time instant t_h is approximated as follows:

$$X'(t_h) = \frac{X(t_h) - X(t_{h-1})}{t_h - t_{h-1}} \tag{3}$$

The computation of the derivatives of order $d > 1$ will follow the same procedure, i.e., first, the raw data of the d-th derivative is computed using the raw data of the $(d-1)$-h derivative, to be then smoothed with an appropriate interpolation technique.

Hence, we choose to apply the smoothing step after taking the numerical derivatives. An alternative option is to first smooth the function and then calculate the derivatives. Indeed both choices are possible. However, since, as indicated in Section 4, we are using a cubic spline as an interpolation technique, if smoothing is done first and the derivatives are estimated later, then the evaluation of derivatives of degree greater than three will always be zero. Nevertheless, note that in our numerical experience (Section 4), we have fixed the maximum degree of the derivatives to be used to $d = 2$ since the first two derivatives are the most commonly used; therefore, any of the two choices could be applied. An appropriate value of d could be based on domain knowledge or on a model selection step to be embedded in our methodology. This issue is beyond the scope of this paper and may require further investigation.

2.2. Support Vector Regression and kernel definition

The basic idea of the nonlinear tool SVR is to find a function $\hat{Y} : \mathcal{X} \to \mathbb{R}$, in such a way that, for $X \in \mathcal{X}_i$, $\hat{Y}(X_i)$ differs at most from ε from the actually obtained response $Y_i \in \mathbb{R}$. Since this constraint does not usually hold, it is relaxed by introducing slack variables, yielding an optimization problem that minimizes the ε-insensitive loss function and adding a linear penalization of the deviations from ε. Problem (4) is the dual formulation of the resulting optimization problem:

$$\begin{cases} \max\limits_{v,v^*} & -\frac{1}{2}\sum\limits_{i,j\in s}\left(v_i - v_i^*\right)\left(v_j - v_j^*\right)CK(X_i, X_j) - \\ & -\varepsilon\sum\limits_{i\in s}\left(v_i + v_i^*\right) + \sum\limits_{i\in s} Y_i\left(v_i - v_i^*\right) \\ \text{s.t.} & \sum\limits_{i\in s}\left(v_i - v_i^*\right) = 0 \\ & v_i, v_i^* \in [0,1],\ i \in s \end{cases} \tag{4}$$

and the function $\hat{Y}$ can be expressed as:

$$\hat{Y}(X) = \sum_{i \in s} (v_i - v_i^*) CK(X_i, X) + b, \quad X \in \mathcal{X} \tag{5}$$

In Eq. (5), the term b denotes a threshold value which can be easily computed by exploiting the Karush–Kuhn–Tucker (KKT) conditions of Problem (4). See Smola and Schölkopf (2004) for more details. Furthermore, C is a scalar regularization parameter that penalizes the instances whose deviations are larger than ε. Both ε and C are parameters which are usually tuned by using a grid search on a sufficiently large interval. See Ferraty et al. (2010) and Hernández et al. (2007) for a deeper analysis. Finally, K denotes the so-called kernel function. A wide variety of kernels, mostly in finite dimensional spaces, are proposed in the literature. We can mention for instance the linear kernel, Carrizosa and Romero Morales (2013), Cristianini and Shawe-Taylor (2000) and Hofmann et al. (2008), the polynomial kernel, Muñoz and González (2010) and Rossi and Villa (2006), or the Gaussian (RBF) kernel, Carrizosa et al. (2014), Cristianini and Shawe-Taylor (2000) and Keerthi and Lin (2003).

Given a set of H time intervals, $\mathbf{t} = ((t_1, t_2), (t_3, t_4), \ldots, (t_{2H-3}, t_{2H-2}), (t_{2H-1}, t_{2H}))$ in $[0, T]$, where the h-th interval has the form (t_{2h-1}, t_{2h}), for $h = 1, \ldots, H$, we can define, for the p-variate functional data $X = (X_1, \ldots, X_p) \in \mathcal{X}$, a functional kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Particularly, for the Gaussian kernel with bandwidth $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)$ and $X_i, X_j \in \mathcal{X}$, we propose:

$$K(X_i, X_j, \boldsymbol{\omega}, \mathbf{t}) = \exp\left(-\sum_{v=1}^{p} \omega_v \sum_{h=1}^{H} \frac{1}{t_{2h} - t_{2h-1}} \int_{t_{2h-1}}^{t_{2h}} \left(X_{iv}(t) - X_{jv}(t)\right)^2 dt \right). \tag{6}$$

In this paper, we will just focus on the Gaussian kernel due to its well-known effectiveness, though the application to other kernels is straightforward.

It is important to remark that if the length of the interval (t_{2h-1}, t_{2h}) tends to zero, that is to say, the interval tends to a single time instant, namely t_{2h-1}, then, thanks to the integral form of the mean value theorem, Comenetz (2002), the expression of the kernel in (6) tends to

$$K(X_i, X_j, \boldsymbol{\omega}, \mathbf{t}) = \exp\left(-\sum_{v=1}^{p} \omega_v \sum_{h=1}^{H} \left(X_{iv}(t_{2h-1}) - X_{jv}(t_{2h-1})\right)^2 \right), \quad X_i, X_j \in \mathcal{X}, \tag{7}$$

which exactly coincides with the kernel expression used in Blanquero et al. (2019b) for the time instants selection problem.

3. The penalized interval selection problem

After having defined a kernel suitable for intervals and instants and we can proceed with the problem formulation. This section is organized as follows. Section 3.1 explains and organizes the different parameters, coefficients and decision variables involved in the optimization. Afterward, in Section 3.2 the problem of time interval selection in SVR with functional data is formulated. A resolution strategy is proposed in Section 3.3 which will be improved in Section 3.4 by making use of the hierarchical structure of the decision variables in the optimization problem. Finally, Section 3.5 describes how to obtain the best number of time intervals, H_{opt}, in terms of prediction.

3.1. Types of parameters

The aim of this paper is to find a subset of time intervals $(t_1, t_2), \ldots, (t_{2H-1}, t_{2H})$ in such a way that the relationship between the functional predictor X and the scalar response Y, obtained via the SVR problem, is *as good as possible*, in some sense to be specified. Moreover, for the sake of interpretability, we aim to find not very large intervals. Therefore, we penalize large intervals. Such penalization is weighted according to a non-negative parameter λ.

Three very different types of parameters (decision variables) can be found in the time interval selection problem. First, the H time intervals represented by $\mathbf{t} = ((t_1, t_2), (t_3, t_4), \ldots, (t_{2H-3}, t_{2H-2}), (t_{2H-1}, t_{2H}))$ satisfying that $0 \leqslant t_1 \leqslant t_2 \leqslant \cdots \leqslant t_{2H-1} \leqslant t_{2H} \leqslant T$, and second, the regularization parameter λ and the parameters $\varepsilon, C, \boldsymbol{\omega}$ involved in the SVR problem Eq. (4), and in the Gaussian kernel (6). Finally, the third type of decision variables are the coefficients v and v^* in the expression of the response (5), which are the solution of the optimal problem (4).

The traditional SVR approach, in absence of variable selection, has parameters ε, C, and (for the RBF kernel) one scalar parameter, σ. Coefficients v and v^* are found by solving (4) for every combination of ε, C, and σ in a grid. This is doable in the traditional SVR since there are only three parameters to tune. When the number of parameters is large, the computational burden increases exponentially, making it impossible to obtain a solution with such a grid search. Variable-dependent scaling parameter σ, playing the role of our $\boldsymbol{\omega}$ parameters, has been considered in the literature, Carrizosa et al. (2014) and Phienthrakul and Kijsirikul (2005). In the context of FDA, a functional scaling or bandwidth has also been considered, Blanquero et al. (2019a). The increase in the number of parameters makes grid search impractical and unreliable, hence calling for bespoke optimization-based approaches to tuning, Carrizosa et al. (2014), Chapelle et al. (2002) and Friedrichs and Igel (2005).

Note that determining the best parameters via optimization approaches is not a trivial task either: optimization problems are, in general, difficult to solve. The objective function is not explicitly described and is time-consuming to evaluate it. Moreover, the problem lacks gradient-based properties that could potentially guide the optimization process. All these difficulties call for a bespoke approach that captures the structure of the problem as much as possible.

In order to make the tuning problem tractable, we follow the strategy of Blanquero et al., 2019a, 2019b; Jiménez-Cordero and Maldonado, 2020 where a hybrid approach is proposed: grid search + optimization, with an objective function written in analytic form, using a more tractable surrogate of the accuracy. This approach yields a trade-off between the computational burden of the parameter tuning and its performance.

3.2. Problem formulation

Finding the parameters of the SVR can be seen as a bilevel problem. The internal problem is the optimization of v, v^* in (4), which is a quadratic concave maximization problem with linear constraints. The external problem seeks ε, C and σ minimizing the mean squared error on a validation set. Due to the intractability of this external problem, it is approached by a grid search. When translating this approach for the time interval and instant selection case, even the grid-search becomes intractable and a bespoke approach is necessary. Moreover, since we are interested in selecting *short* intervals, a term penalizing the interval lengths will be added to the sum of the squared residuals on the objective function of the external problem:

$$\sum_{i \in s_2} \left(Y_i - \hat{Y}(X_i(\mathbf{t}), C, \boldsymbol{\omega}, v, v^*) \right)^2 + \lambda \sum_{h=1}^{H} (t_{2h} - t_{2h-1})^2. \tag{8}$$

The expression $\hat{Y}(X_i(\mathbf{t}), C, \boldsymbol{\omega}, v, v^*)$ represents the predicted value. The dependence on the time instants of $\mathbf{t}$ is emphasized in the predicted value through the functional datum $X_i(\mathbf{t})$. Note that ε is missing from that expression because the residuals depend

on it only through the SVR coefficients v, v^*, i.e., via the optimal solution of the inner Problem (4). Observe that our problem has a much higher number of decision variables than the traditional approach. Fortunately, we can take advantage of the functional nature of the datum X_i and the continuous dependence of $C, \boldsymbol{\omega}$ and $\mathbf{t}$ on (8). The second term in (8) is a regularization term that penalizes the length of the interval. The value of λ measures the trade-off between the sum of the squared residuals and the length of the intervals. Large values of λ yield the evaluation of the sum of the squared residuals at tiny intervals, which may become time instants in the degenerate case, i.e., $\lambda = +\infty$. The other extreme case, $\lambda = 0$ just seeks for the minimum sum of the squared residual values according to the information given by the intervals of any length.

Optimizing λ in expression (8) makes no sense, since it is a regularization parameter that trades off the residuals versus the length of the chosen intervals. In the same way, it makes no sense to optimize ε in Problem (4). This calls for a three level approach, where ε and λ are sought by a grid search, and the remaining parameters and coefficients are optimized in a bilevel problem where the outer problem optimizes $C, \boldsymbol{\omega}$ and $\mathbf{t}$ on (8) and the inner problem optimizes v, v^* in Problem (4).

Before finally presenting the formulation of this bilevel problem, we will introduce another innovation. In the traditional approach, in order to obtain more stable results and avoid overfitting, the complete set of individuals is usually divided in the literature into three subsets, namely, training, validation, and testing, with k-fold cross-validation as the model selection criterion. These samples are used to train the model, get the best parameters and estimate the final accuracy, respectively. In this paper, we take this idea further, and we divide the sample s into four independent samples, namely s_1, s_2, s_3 and s_4. These samples are obtained as follows. We first divide the sample into k folds. Then, $k-1$ folds are randomly selected and divided into three parts of equal size, yielding samples s_1, s_2 and s_3. Finally, the remaining fold forms sample s_4. Samples s_1 and s_2 play the role of training samples, whilst s_3 and s_4 are the validation and testing samples, respectively. Particularly, the independent sample s_1 is used to obtain the optimal values of v and v^* by solving Problem (4) for fixed $C, \boldsymbol{\omega}, \mathbf{t}$ and ε. Sample s_2 is employed to compute the sum of the squared residuals between the response Y_i and the predicted response value $\hat{Y}(X_i(\mathbf{t}), C, \boldsymbol{\omega}, v, v^*)$, and the regularization term weighted by λ. Sample s_3 is utilized to tune the parameters (λ, ε), by evaluating the sum of the squared residuals in the grid, and keeping the parameter yielding the smallest value. Finally, sample s_4 is used to estimate the sum of the squared residuals and test the final results. Even though the success of the SVR model is highly dependent on the training data set size, we observed in initial experiments that, if the same training set is used for the SVR and the least squares optimization, i.e., $s_1 = s_2$, then overfitting appears. This is the reason why the complete set of individuals is divided into four subsets.

Hence, for a given pair (λ, ε), the bilevel optimization problem is stated as follows:

$$\begin{cases} \min\limits_{C,\boldsymbol{\omega},\mathbf{t},v,v^*} & \sum\limits_{i\in s_2}\left(Y_i - \hat{Y}(X_i(\mathbf{t}), C, \boldsymbol{\omega}, v, v^*)\right)^2 + \lambda\sum\limits_{h=1}^{H}(t_{2h} - t_{2h-1})^2 \\ \text{s.t.} & v, v^* \text{ solves (4) in } s_1, \\ & C \geqslant 0, \\ & \omega_v \geqslant 0, \forall v \\ & 0 \leqslant t_1 \leqslant \cdots \leqslant t_{2H} \leqslant T \end{cases} \tag{9}$$

Note that additional constraints over the time intervals can be easily added to the previous problem. For instance, one can impose that two consecutive time intervals are separated by at least a fixed distance.

3.3. An alternating algorithm

Problem (9) can be handled with off-the-shelf bilevel optimization techniques, such as Colson et al. (2007). Such tools are computationally very expensive, and thus we propose, instead, an alternating approach, as also done in Blanquero et al., 2019a, 2019b; Jiménez-Cordero and Maldonado, 2020.

In the first step of our alternating procedure, Problem (4) is solved in sample s_1 for given $C, \boldsymbol{\omega}$ and $\mathbf{t}$, obtaining the optimal SVR variables v and v^*. Problem Eq. (4) is a quadratic concave maximization problem with linear constraints. Hence, classic local search routines may be applied to find the global optimum.

In the second step, for v and v^* fixed, we obtain the optimal values of $C, \boldsymbol{\omega}$ and $\mathbf{t}$ solving Problem Eq. (10) in sample s_2:

$$\begin{cases} \min\limits_{C,\boldsymbol{\omega},\mathbf{t}} & \sum\limits_{i\in s_2}\left(Y_i - \hat{Y}(X_i(\mathbf{t}), C, \boldsymbol{\omega})\right)^2 + \lambda\sum\limits_{h=1}^{H}(t_{2h} - t_{2h-1})^2 \\ \text{s.t.} & C \geqslant 0, \\ & \omega_v \geqslant 0, \forall v \\ & 0 \leqslant t_1 \leqslant \cdots \leqslant t_{2H} \leqslant T \end{cases} \tag{10}$$

Thanks to the functional nature of the data, Problem (10) is a continuous optimization problem which is solved by combining standard local searches with a multistart strategy to avoid getting stuck at poor local optima.

The alternating approach is run, for a fixed pair (λ, ε), until some stopping criteria is met, yielding suitable values of $C, \boldsymbol{\omega}, \mathbf{t}, v$ and v^*. The value of (λ, ε) is obtained with a grid search, i.e., computing, for each (λ, ε) in a grid, the sum of the squared residuals on the sample s_3, and keeping those associated to the smallest value. Finally, to test the efficiency of our approach, we calculate the sum of the squared residuals in a fourth independent sample s_4.

The pseudocode of our proposal can be seen in Algorithm 1, and an improvement in the solving strategy is proposed in Section 3.4.

Note that the proposed alternating method forces us to split the data set into four samples $s_1, \ldots, s_4$ but this is a simple and effective strategy (according to our numerical results) to handle the challenging bilevel optimization problem formulated for tuning the SVR parameters. Nevertheless, this strategy may be computationally expensive or unstable, especially in the cases of small-size data sets where the subsamples will be even smaller. Therefore, even though theoretically, our approach can also be applied when no feature selection is performed, we believe that the alternating procedure is not advisable to train an SVR problem when no feature selection is carried out, but just as a strategy to deal with the multiple tuning parameters problem as done in this manuscript. A deeper analysis of other existing tuning methods on this topic deserves further study. For further information on the existing strategies, the reader is referred to Kaneko and Funatsu (2015), Smets et al. (2007) and Sreekumar and Verma (2016).

Algorithm 1 Heuristic for variable selection

Input: H.
- Randomly split the sample s into s_1, s_2, s_3 and s_4.
- Compute the derivatives of the functional data.
- Smooth the data with some interpolation technique.

for (λ, ε) in the grid **do**
  Alternating Procedure

(*continued on next page*)

repeat

1. For $C, \boldsymbol{\omega}, \mathbf{t}$ fixed, calculate the parameters v, v^* of the SVR problem (4) using the sample s_1.
2. Fixed v and v^* fixed, calculate $C, \boldsymbol{\omega}, \mathbf{t}$ by solving Problem (10) over sample s_2.

until stopping criteria

- Evaluate the sum of the squared residual values using the sample s_3

for (λ, ε) fixed in the grid.

end for

- The optimal value of (λ, ε) are those with minimum sum of the squared residual value in s_3, and the optimal values of v, v^*, C, $\boldsymbol{\omega}$ and $\mathbf{t}$ are the parameters associated to the optimal (λ, ε).

Output: Optimal parameters $\lambda, \varepsilon, C, \boldsymbol{\omega}, \mathbf{t}, v, v^*$, and the sum of the squared residual estimated on s_4.

3.4. A nested heuristic

Algorithm 1 may get stuck in poor local minima. In order to avoid an unmanageable number of local searches when the number of time intervals increases, we enhance the heuristic proposed in Section 3.3. More specifically, based on the works (Blanquero et al., 2019a; Carrizosa et al., 2014), we propose to define as in Blanquero et al. (2019b) a series of nested models of increasing complexity, in which the optimal solution of a simple model is employed as initial solution in a more complex case. In other words, when seeking the $h+1$ time intervals in $\mathbf{t}^{h+1}$, one uses as initial solution a perturbation of $\mathbf{t}^h$, i.e., the optimal solution obtained when only h time intervals are sought. Particularly, the initial solution of the parameters C and $\boldsymbol{\omega}$ in the level $h+1$ are set as the optimal solution of such parameters in the level h, $\boldsymbol{\omega}^h_{opt}$ and C^h_{opt}, respectively. Moreover, the choice of the initial solution of the $h+1$ time intervals in $\mathbf{t}^{h+1}$ is made by selecting random values $(\tau_1, \tau_2) \in [0,T] \times [0,T]$, and including it in the appropriate position of the optimal solution of the level h, $\mathbf{t}^h_{opt}$. In other words, $\mathbf{t}^{h+1}_{opt} = \sigma\left(\tau_1, \tau_2, \mathbf{t}^h_{opt}\right)$, where σ is a function that sorts in increasing order the time instants in $\mathbf{t}^h_{opt}$ together with τ_1 and τ_2.

The pseudocode of the nested heuristic is outlined in Algorithm 2.

Algorithm 2 Nested heuristic for variable selection

Input: H, nested kernels $K(X_i, X_j, \boldsymbol{\omega}^1, \mathbf{t}^1) \prec \ldots \prec K(X_i, X_j, \boldsymbol{\omega}^H, \mathbf{t}^H)$.

- Randomly split the sample s into s_1, s_2, s_3 and s_4.
- Compute the derivatives of the functional data.
- Smooth the data with some interpolation technique.

for (λ, ε) in the grid

Initialization:

- $h := 1$.
- Randomly select an initial solution $\widetilde{C}^1 \in [0, +\infty)$, $\tilde{\boldsymbol{\omega}}^1 \in [0, +\infty)^p$ and $\tilde{\mathbf{t}}^1 := (t_1, t_2) \in [0,T] \times [t_1, T]$.
- Set $(C, \boldsymbol{\omega}, \mathbf{t}) := \left(\widetilde{C}^1, \tilde{\boldsymbol{\omega}}^1, \tilde{\mathbf{t}}^1\right)$.

while $h \leqslant H$

1. Run the Alternating Procedure of Algorithm 1 for $K\left(X_i, X_j, \boldsymbol{\omega}^h, \mathbf{t}^h\right)$, starting from $(C, \boldsymbol{\omega}, \mathbf{t})$ and yielding $\left(C^h_{opt}, \boldsymbol{\omega}^h_{opt}, \mathbf{t}^h_{opt}\right)$ as solution, using samples s_1 and s_2.
2. Randomly generate $(\tau_1, \tau_2) \in [0,T] \times [0,T]$.
3. Set $C^{h+1} := C^h_{opt}$, $\boldsymbol{\omega}^{h+1} := \boldsymbol{\omega}^h_{opt}$, $\mathbf{t}^{h+1} := \sigma\left(\tau_1, \tau_2, \mathbf{t}^h_{opt}\right)$, $(C, \boldsymbol{\omega}, \mathbf{t}) := \left(C^{h+1}, \boldsymbol{\omega}^{h+1}, \mathbf{t}^{h+1}\right)$ and $h := h+1$.
4. Evaluate the sum of the squared residuals over the sample s_3 with (λ, ε) fixed.

end while

end for

- For each h, the optimal value of (λ, ε) is the one with the minimum sum of the squared residual in s_3. The optimal values of v, v^*, C, $\boldsymbol{\omega}$ and $\mathbf{t}$ are the parameters associated to the optimal (λ, ε).

Output: Optimal parameters C^h_{opt}, $\boldsymbol{\omega}^h_{opt}$, $\mathbf{t}^h_{opt}$, $\forall h$, the associated coefficients $\lambda, \varepsilon, v, v^*$, and the sum of the squared residual estimated on s_4.

Observe that our proposed heuristic differs from a greedy approach (Berrendero et al., 2019; Ferraty et al., 2010), since our proposal utilizes the optimal solution of level h as a starting solution of the level $h+1$, allowing a very different solution for level $h+1$ than the one obtained in the previous level, h. Consequently, our approach gives more flexibility to the model.

Moreover, when the exact number of selected time intervals, H, is to be determined, our algorithm has the advantage of allowing us to build a trajectory of the sum of the squared residuals in terms of the number of time intervals. It may be very useful when a list of models with different complexity is needed.

3.5. Estimating the optimal number of variables, H_{opt}

In our methodology, we have proposed to build a trajectory of the sum of the squared residuals according to the prefixed number of time intervals h, going from $h=1$ to $h=H$. It is clear that the solution so-obtained is highly dependent on the value of h. In fact, smaller values of h produce more interpretable models, since a small number of time intervals is to be kept into account. However, the regression is more accurate (in the training sample) if a large number of time intervals is sought. It is then necessary to follow a criterion that allows us to get a trade-off between accuracy and interpretability.

In the literature, one can find custom strategies for tuning the parameter defining the number of time intervals. This is the case of Berrendero et al. (2019), in which change point detection methods based on k-means algorithms are applied, or Ferraty et al. (2010), where the number of design points is automatically found with their proposed forward–backward strategy. However, most works tune this parameter with standard model selection approaches, such as the Bayesian Information Criterion (BIC), Schwarz (1978), used in Kneip et al. (2016) or Grollemund et al. (2019).

By contrast, in this paper, we propose to select the parameter H_{opt} using cross-validation. More precisely, the sum of the squared residuals are measured on the sample s_3 for all possible values of h ranging from 1 to H, and we keep the parameter with the smallest value.

We acknowledge the fact that the proposed procedure can be used to also determine d, ε, and λ, and consequently, the associated computational cost will be enlarged. When computational time is an issue, the authors recommend a straightforward parallelization of the algorithm.

4. Experimental setup

This section describes the experimental setup on the proposed models. Section 4.1 presents the experiments performed and Section 4.2 details the data sets used to test our methodology.

4.1. Description of the experiments

Algorithm 2 is run to show the usefulness of our approach, i.e., to test whether the predictions obtained when H time intervals are carefully chosen are comparable to, or even better than, the sum of the squared residuals achieved when the full time interval is considered.

As a preprocessing step, the data and the subsequent derivatives have been smoothed using the procedure explained in Section 2.1. For the sake of simplicity, the well-known cubic spline interpolation technique, De Boor (1978) has been used to smooth the data. Other interpolation techniques, such as B-splines, De Boor (1978) can be also applied without damaging the results.

Three different experiments are performed in this paper. First, Algorithm 2 is run with $\lambda = 0$ in the objective function of Problem (9), i.e., we seek the most informative time intervals without paying attention to their length. Second, Algorithm (9) is executed with $\lambda = +\infty$, which is equivalent to looking for intervals of zero length, that is to say, time instants. Last but not least, we run Algorithm (9) for λ in the grid $\left\{10^{-2}, \ldots, 10^{2}\right\}$ in logarithmic scale.

To get stable out-of-sample results, k-fold cross-validation is used in the three experiments described above. In the literature, k is often chosen as 10. However, when the data set is very small, $k = 10$ is not large enough to get stable results, and *leave-one-out* cross-validation is preferred, i.e., k coincides with the number of observations. Our selection of the number of folds has been made dependant on the size of the data sets. More precisely, if a database is big, then $k = 10$ is chosen. By contrast, in the small data sets, leave-one-out is performed. Here, we consider that a database is big if it has more than 100 observations. More details about the cardinality of the databases can be seen in Table 1. Algorithm 2 is run k times, one per fold. Each time, the data set is split into four parts, $s_1 - s_4$, as described in Section 3.2. As the output of our methodology, we provide the average sum of the squared residuals estimated on the test sample s_4 over all the folds in the cases where $\lambda = 0, \lambda = +\infty$, and $\lambda \in \left\{10^{-2}, \ldots, 10^{2}\right\}$.

The number of runs in the multistart strategy applied in solving Problem 10 is five. The Alternating Procedure stops either when ten iterations are executed or when the difference between the sum of the squared residual values of two consecutive iterations is less than 10^{-5}. The maximum number of time intervals to be sought is $H = 10$, and the parameter ε moves in the set $\left\{10^{-8}, \ldots, 10^{-1}\right\}$ in logarithmic scale.

All the experiments are carried out on a cluster with 2 Tb of RAM memory at 6.2 TFlops, running CentOS Linux 7.3, and it is coded in R, Core Team (2017).

Table 1
Data description summary.

	#records	#time instants	#components
FHV	1500	100	1
MK005	300	101	3
MK01	300	101	3
PSVone	100	200	1
PSVthree	100	300	1
canadian	35	365	1
cookie	72	700	1
DTI	334	93	1
gasoline	60	401	1
marzipan_moisture	32	600	1
marzipan_sugar	32	600	1
sugar	268	571	1
sunflower	111	309	1
tecator	215	100	1

4.2. Description of the data sets

We have tested our proposal on 12 univariate and two multivariate databases, widely used in the literature on functional regression. Five of these data sets are simulated according to models available in the literature: three univariate functional data model (namely *FHV*, Ferraty et al. (2010); and *MK005* and MK01, Matsui and Konishi (2011)) and two multivariate functional data models (named PSVone and PSVthree, Picheny et al. (2019)). Nine data sets contain data from real applications: *canadian*, Goldsmith and Scheipl (2014), James et al. (2009) and Tutz and Gertheiss (2010), *cookie*, Goldsmith and Scheipl (2014), *DTI*, Goldsmith and Scheipl (2014), *gasoline*, Park et al. (2016), *marzipan_moisture*, Tutz and Gertheiss (2010), *marzipan_sugar*, Tutz and Gertheiss (2010), *sugar*, Aneiros and Vieu (2014), *sunflower*, Picheny et al. (2019) and *tecator*, Ferraty et al. (2010), Goldsmith and Scheipl (2014) and Picheny et al. (2019). The data sets *marzipan_moisture* and *marzipan_sugar* share the same independent (functional) variables and only differ on the response variable. The same happens with the simulated data sets pair *MK005* and *MK01*, and also with the pair *PSVone* and *PSVthree*.

To give an idea of the variety of functions we are dealing with, a sample from ten individuals of each database is given in Fig. 1. Table 1 provides a summarized description of the data sets, including the number of records, the number of time instants where data are discretized, and the number of covariates. Details about each data set are presented in the following subsections.

4.2.1. FHV data set

According to the example of Section 3.2 of Ferraty et al. (2010) we have generated 1500 curves discretized in 100 equispaced points in the interval $[0, 2\pi]$ following this structure:

$$X_i(t) = \sum_{\ell=1}^{3} U_{i\ell} \cos\{(3+\ell)t\} + \sum_{\ell=1}^{3} V_{i\ell} \sin\{(4+\ell)t\} + W_i(t-\pi)^2, \quad i = 1, \ldots, 1500$$

where $U_{i\ell}, V_{i\ell}, W_i, \ell = 1, 2, \forall i$ are uniformly distributed in $[0, 1]$, whereas U_{i3} and V_{i3} follow a normal distribution $\mathcal{N}(0, 0.25)$. For $i = 1, \ldots, 1500$, the values of the response variable are obtained from the model $Y_i = r(X_i) + \gamma_i$, based on the time instants $t \in \left\{\frac{48\pi}{99}, \frac{58\pi}{99}, \frac{128\pi}{99}\right\}$ with

$$r(X_i) = X_i\left(\frac{48\pi}{99}\right) + 2X_i\left(\frac{58\pi}{99}\right)X_i\left(\frac{128\pi}{99}\right) \tag{11}$$

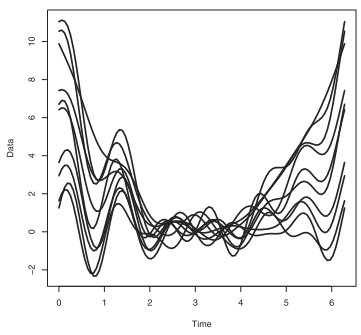
and γ_i independent and identically distributed as $\mathcal{N}\left(0, \sigma_\gamma^2\right)$, with $\sigma_\gamma^2 = 5\%\, var\{r(X_i)\}$.

4.2.2. MK005 and MK01 data sets

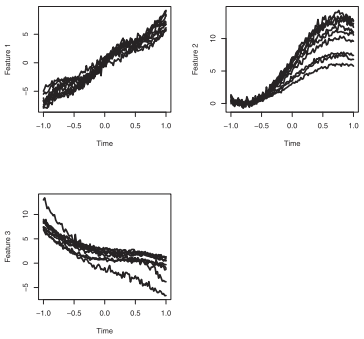
In this section we have worked with 300 observations of two data sets, namely *MK005* and *MK01* generated according to Matsui and Konishi (2011). The three predictor variables are built as follows for $t \in [-1, 1]$:

$$X_v(t) = U_v(t) + \gamma_v, \quad v = 1, 2, 3, \tag{12}$$

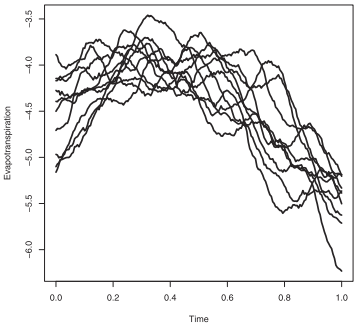
where $\gamma_v \sim \mathcal{N}\left(0, 0.025 \cdot (r_v)^2\right)$ and $r_v = \max_t(U_v(t)) - \min_t(U_v(t))$. Furthermore,



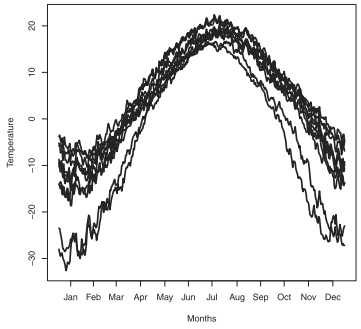
(a) FHV



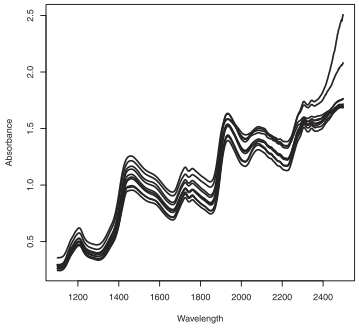
(b) MK005 and MK01



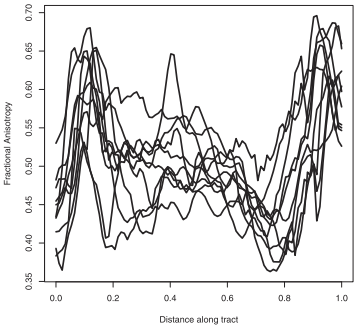
(c) PSVone and PSVthree



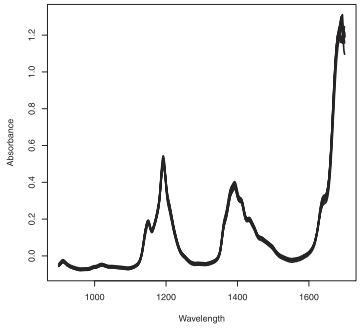
(d) canadian



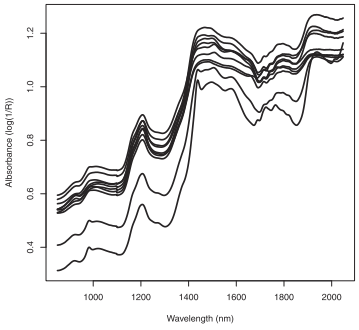
(e) cookie



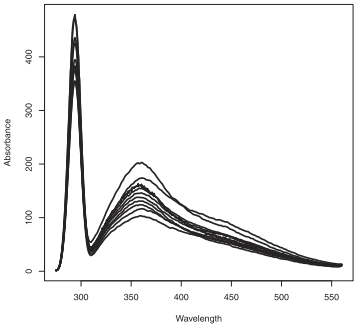
(f) DTI



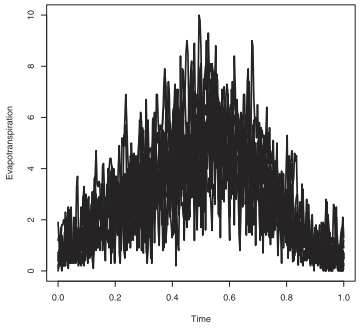
(g) gasoline



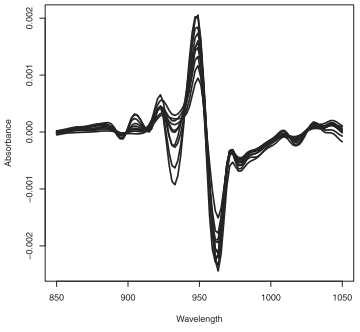
(h) marzipan_moisture and marzipan_sugar



(i) sugar



(j) sunflower



(k) tecator

Fig. 1. Sample of ten observations of the databases.

$$\begin{array}{lll} U_1(t)=\cos(2\pi(t-a_1))+a_2 t; & a_1\sim\mathcal{N}\left(-5,3^2\right); & a_2\sim\mathcal{N}(7,1) \\ U_2(t)=b_1\sin(2t)+b_2; & b_1\sim\mathcal{U}(3,7); & b_2\sim\mathcal{N}(0,1) \\ U_3(t)=c_1t^3+c_2t^2+c_3t+c_4; & c_1\sim\mathcal{N}\left(-3,1.2^2\right); & c_2\sim\mathcal{N}\left(2,0.5^2\right); \\ & c_3\sim\mathcal{N}(-2,1); & c_4\sim\mathcal{N}\left(2,1.5^2\right) \end{array} \tag{13}$$

The response variable Y is built according to the following rule:

$$Y=g(U)+\xi \tag{14}$$

with

$$\begin{array}{lll} g(U)=\sum_{v=1}^{3}\int_{-1}^{1}U_v(t)\beta_v(t)dt; & \xi\sim\mathcal{N}\left(0,(c\cdot s)^2\right); & s=\max(g(U))-\min(g(U)) \\ \beta_1(t)=\sin(2\pi t); & \beta_2(t)=\sin(\pi t); & \beta_3(t)=0 \end{array} \tag{15}$$

The value of c in the probability distribution of the parameter ξ in Eq. (15) depends on the multivariate data set, *MK005* or *MK01*, we are dealing with. More precisely, $c=0.05$ in the database *MK005* and $c=0.1$ in the case of *MK01*.

4.2.3. PSVone and PSVthree data sets

We have generated these two databases according to the example in Section 6.1 of Picheny et al. (2019) and using the scripts available at https://github.com/tuxette/appliSISIR/tree/master/RLib. More specifically, 100 curves at 200 and 300 evaluation points in databases *PSVone* and *PSVthree*, respectively in the interval $[0,1]$ have been created. For the sake of simplicity, we have unified the definition domain of the three functional variables involved in this example. The expression of the independent variable is given by:

$$X(t)=U(t)+\gamma \tag{16}$$

where $U(t)$ is a Gaussian process indexed on $[0,1]$ with mean $\mu(t)=-5+4t-4t^2$, and the Matern 3/2 covariance function. The parameter γ is a centered Gaussian variable independent on U.

The response variable Y has the form:

$$Y=\sum_{\ell=1}^{L}\log|\langle X,\beta_\ell\rangle| \tag{17}$$

with

$$\beta_\ell(t)=\sin\left(\frac{t(2+\ell)\pi}{2}-\frac{(\ell-1)\pi}{3}\right)\mathbb{I}_{I_\ell}(t) \tag{18}$$

and $\mathbb{I}_{I_\ell}(t)$ is the indicator function which takes the value 1 at I_ℓ and 0 otherwise.

Particularly, the values of L and I_ℓ depend on the data set we are dealing with. In the data set *PSVone*, $L=1$ and $I_1=[0.2,0.4]$. By contrast, in *PSVthree*, $L=3$ and $I_1=[0,0.1], I_2=[0.5,0.65]$ and $I_3=[0.65,0.78]$.

4.2.4. Real applications data sets

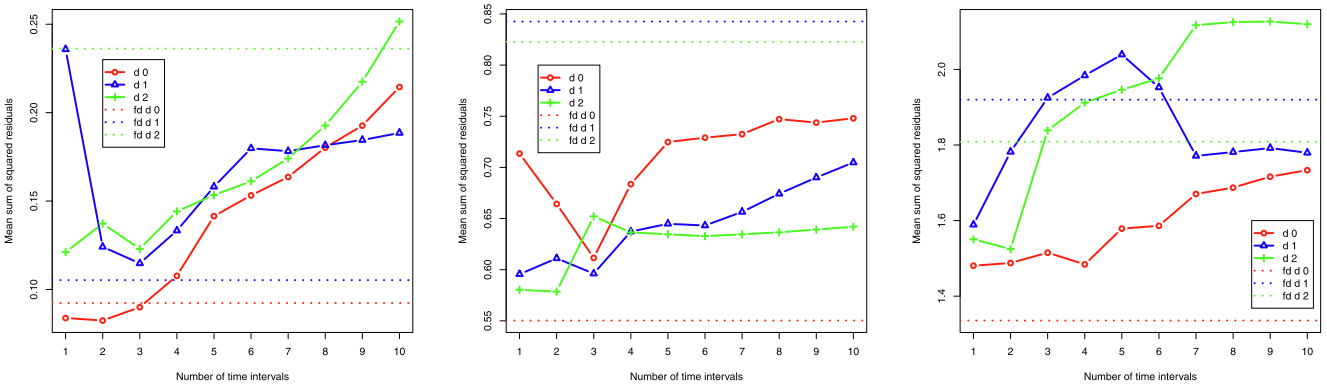
This section presents the details of the real data sets that have been used in the numerical experience:

canadian The *canadian* data set, Goldsmith and Scheipl (2014) and James et al. (2009), is formed by the daily temperature along one year measured on 35 Canadian weather stations. The goal is to predict the logarithm of the total annual rainfall.
cookie The *cookie* database comes from Goldsmith and Scheipl (2014) and measures the 72 spectra of cookie dough samples every two nanometers (nm) from 1100 to 2498 nm with the aim of predicting the percentage of sucrose content.
DTI The *DTI* data set, Goldsmith and Scheipl (2014), consists of 334 observations that measure the white matter in the corpus callosum to predict the cognitive performance in order to study multiple sclerosis lesions.
gasoline This data set is denoted by *gasoline* and comes from Park et al. (2016). It can be obtained from the `R` library `refund`. It is formed by 60 spectra of gasoline measured at 401 equispaced points by diffuse reflectance ranging from 900 nm to 1700 nm in order to predict the octane number.
marzipan The data sets *marzipan_moisture* and *marzipan_sugar* have been used in Tutz and Gertheiss (2010) and can be downloaded from www.models.kvl.dk/Marzipan. They are formed by 32 marzipan spectra measured every two nm from 850 to 2050 nm. The goal is to respectively predict in the databases *marzipan_moisture* and *marzipan_sugar*, the moisture and sugar contents in marzipan.
sugar The goal of data set *sugar*, from Aneiros and Vieu (2014), is to predict the percentage of ash content from the fluorescence spectra, measured on 266 samples of sugar.
sunflower The *sunflower* database comes from Picheny et al. (2019) and can be obtained in https://github.com/tuxette/appliSISIR/tree/master/data. It consists of a set of 111 climate evaportranspiration daily recordings at 309 points. The objective is to predict the annual grain yield.
tecator The data set *tecator* deals with the near-infrared absorbance spectra of 215 samples of finely chopped pork. The response variable represents the fat content. More details can be found in Ferraty et al. (2010), Goldsmith and Scheipl (2014) and Picheny et al. (2019).

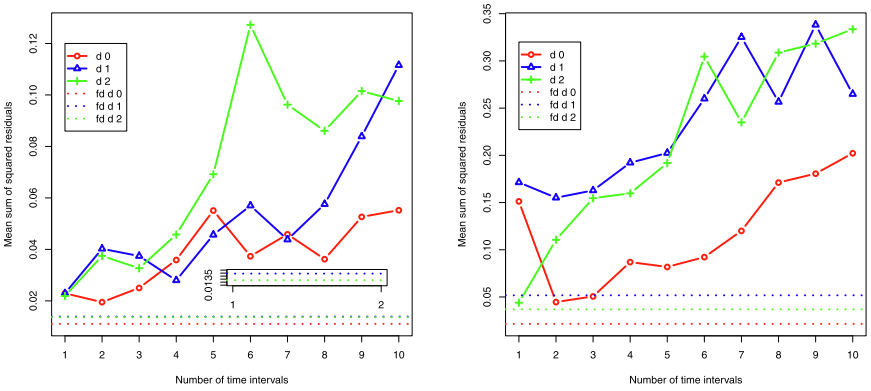
5. Results

In this section we present a numerical evaluation of our approach. For the simulated data sets, Figs. 2, 4, and 9 show respectively, the trajectory of the mean sum of the squared residuals obtained when $\lambda=0,\lambda=+\infty$, and when the best $\lambda\in\left\{10^{-2},\dots 10^{2}\right\}$ is chosen in Problem (9) and h time intervals are sought, ranging from $h=1$ to $h=H$, with $H=10$. Therefore, in the first case, we are seeking intervals of any length, in the second case, we select the most informative time instants, and, in the third case, we are penalizing large intervals. Algorithm 2 is run for three different values of the derivatives $d=0,1,2$, which include, respectively, the situations where just the information of the raw functional data, or their monotonicity ($d=1$, blue), or both their monotonicity and convexity ($d=2$, green) are considered. In the above-mentioned figures, we depict in dotted-red, triangled-blue and crossed-green solid lines the results when $d=0,1,2$ derivatives are considered, respectively. For data based on real-life applications, the same information is shown in Figs. 3, 5, and 10. The exact values of all the averages of the sum of the squared residuals, as well as their standard deviations (in parentheses), are given in Tables 2–4 for comparison purposes. Last column of Tables 2–4 includes the average value of the best value *H_opt* obtained when $h=10$ time intervals (of any length) are sought using the strategy proposed in Section 3.5. As an illustration, Fig. 6 provides the boxplots over all the folds of the best *H_opt* value obtained when running Problem (9) for the database *cookie*, $\lambda=+\infty$, and for the different values of h going from 1 to 10.

In the following subsections, we describe the results in more detail. In particular, Section 5.1 compares our results with those that appear in the literature. Section 5.2 analyzes the results obtained using our approach for selecting time instants or intervals and compares them to the prediction values obtained when the information over the full time domain is considered. Finally, Section 5.3 describes the behavior of our approach with respect to the penalization parameter λ.



(a) FHV (b) MK005 (c) MK01



(d) PSVone (e) PSVthree

Fig. 2. Mean sum of squared errors for $\lambda = 0$, i.e., selecting any length intervals. Results on simulated data.

5.1. Comparison with literature results

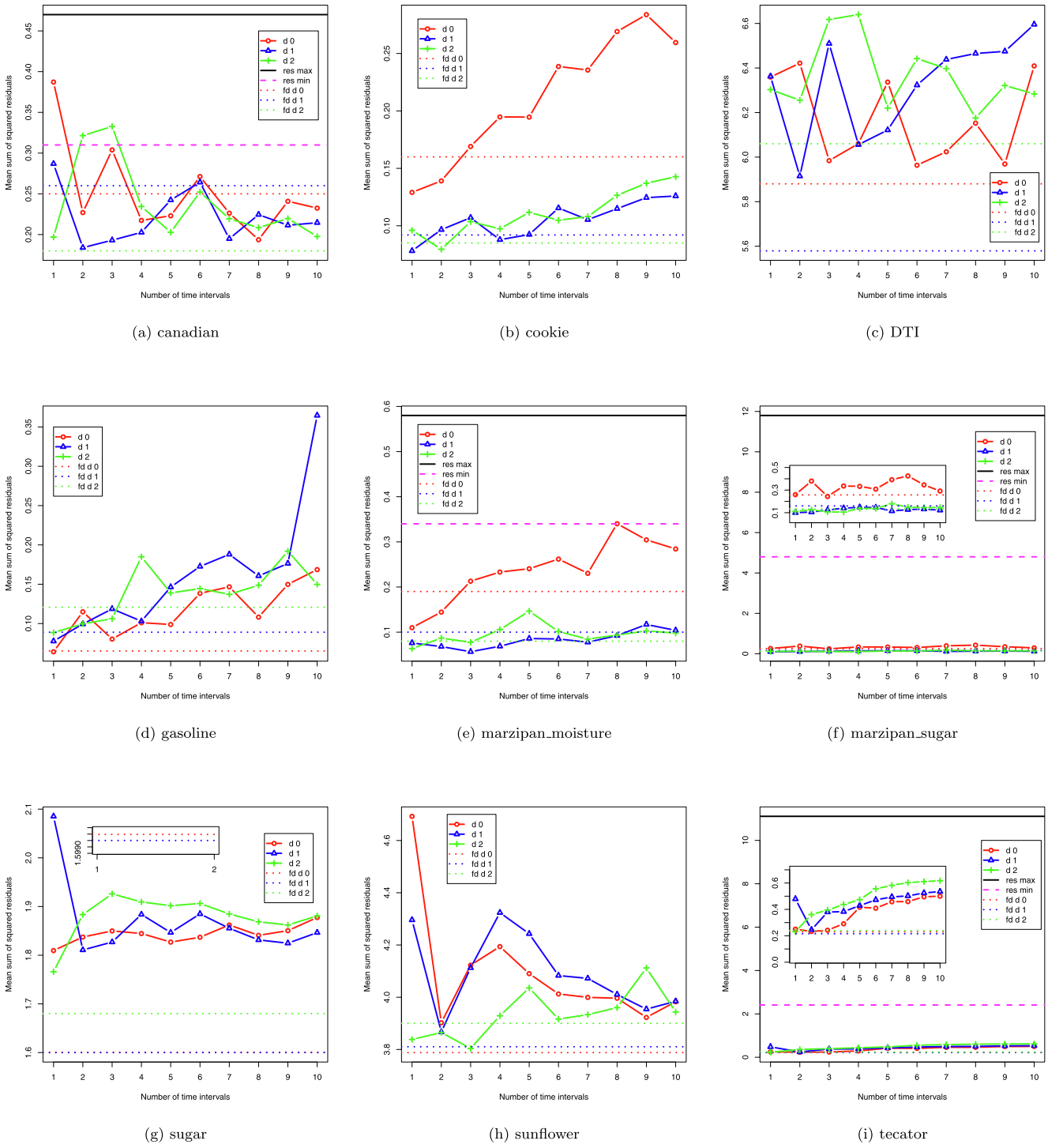
To compare our methodology, we use the maximum (worst) and minimum (best) values of the mean sum of squared errors found in the literature for other interval or instant selection methodologies. We indicate the reference mean sum of squared errors, and their standard deviations just in the cases where they are available: the maximum (worst) mean sum of squared errors in solid black line and the minimum (best) mean sum of squared errors in dashed pink line. In the case where time intervals of any length are sought ($\lambda = 0$), mean reference values for *canadian*, *marzipan_moisture* and *marzipan_sugar* from Tutz and Gertheiss (2010) and *tecator* from Picheny et al. (2019) are available. On the other hand, in the case with $\lambda = +\infty$, that is to say, when just time instants are sought, the mean reference values of the data sets *FHV* and *tecator* from Ferraty et al. (2010), and *sugar* from Aneiros and Vieu (2014), as well as the standard deviation of the data set *FHV*, can be used for the sake of comparison. The case where λ moves in a grid is not considered here, since it has not been handled in the literature, and therefore, the comparison would be unfair.

Note that since the median of the sum of the squared residuals values is given as reference in Ferraty et al. (2010) for the *FHV* database when time instants are sought, in this example, we give as output the median values instead of the mean, as the *y*-axis label of Fig. 4(a) indicates. Table 3 also shows the median values for this database.

It is worth noting that the size of the training and testing sets used in this paper are different from thoses used in the references. However, there is no clear evidence of bias benefiting any method over the others, and therefore the discrepancies that may be in the data set split are meaningless.

Let us now observe that in the case of $\lambda = 0$, for a large enough value of the maximum number of intervals ($h > 3$) and any value of d, our methodology is consistently getting better results than the best available in the literature for that data set. Indeed we have two cases in which our methodology is over 30 times better than the best result available in the literature and another where it is three times better. In the case of $\lambda = +\infty$, i.e., for instants selection, we found results in the literature for three data sets. In the simulated data set *FHV*, when $d = 1$ we are able to consistently outperform the best result in the literature for a high enough value of h (namely $h > 4$). In the other two cases, which correspond to data from real applications, we consistently improve the best result in the literature for any value of d or h. The improvements range from reducing the mean sum of squared errors to a half of the best result in the literature (for *sugar* data set) to a ten-fold reduction for *tecator* data set.

Even taking into account the differences in sample size, the consistency and the order of magnitude of the improvement with respect to the existing literature make us believe that our proposal is able to capture some nonlinearity of the phenomena while existing methods either are only able to deal with the linear case or failing to capture such nonlinearity.



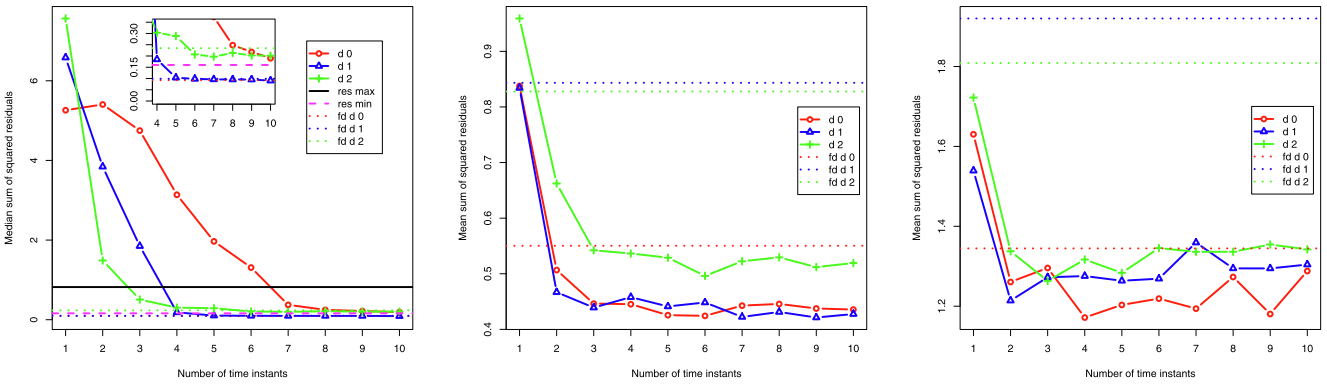
(a) canadian (b) cookie (c) DTI

(d) gasoline (e) marzipan_moisture (f) marzipan_sugar

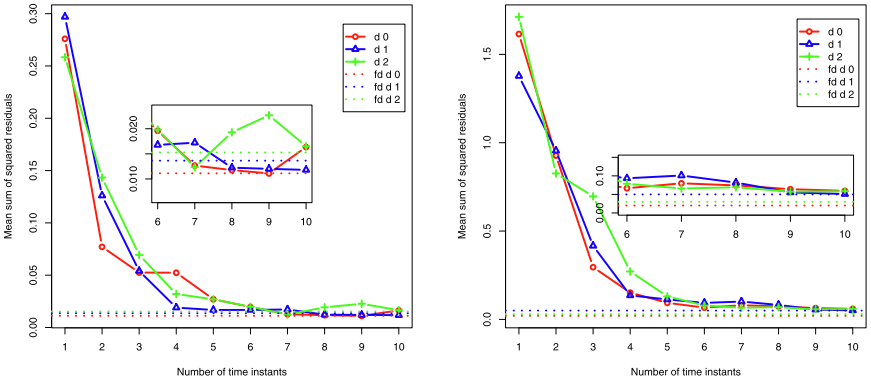
(g) sugar (h) sunflower (i) tecator

Fig. 3. Mean sum of squared errors for $\lambda = 0$, i.e., selecting any length intervals. Results on data from real applications.

Regarding the interpretability of the results, we can highlight for example the data set *canadian*. More precisely, James et al. (2009) asserts that the temperatures in the spring and fall months do have a noticeable effect when predicting the annual rainfall. More specifically, Fig. 4 of James et al. (2009) shows that such an effect is produced around the months of April and November. Fig. 7 shows the density histogram of the time instant values when $h = 3$ time points are sought using our methodology, i.e., with $\lambda = +\infty$. We clearly observe that there exists an evident maximum in the month of November independently of the value of the parameter $d \in \{0, 1, 2\}$. In addition, there is a set of months around April, namely March, April, and May, which are selected with high frequency. Similar results are obtained when $h \neq 3$, and therefore, due to the page limit, no more histogram figures are plotted.



(a) FHV (b) MK005 (c) MK01



(d) PSVone (e) PSVthree

Fig. 4. Mean sum of squared errors for $\lambda = +\infty$, i.e., selecting time instants (intervals of zero length). Results on simulated data.

5.2. Comparison with using the full (time) domain

In order to quantify the contribution of the interval selection with respect to using the whole domain, we have run Algorithm 2 with the same settings as in Section 4.1, i.e., number of iterations, stopping criterion, and values of the parameter ε, to then get the variables v, v^*, C and $\boldsymbol{\omega}$ of Problem (9) together with the average mean sum of squared errors across folds in exactly the same testing sample s_4. The results of Algorithm 2 when no variable selection is performed, i.e., the full (time) domain is taken into account, are plotted on dotted red, blue or green line depending on the degree of the derivatives $d = 0, 1$ or 2, respectively.

When comparing full domain use versus interval selection with no length restrictions (i.e., $\lambda = 0$), Fig. 2 shows that results on three out of five simulated data sets are better than their full domain counterparts when a lower number of intervals is sought and $d = 2$.

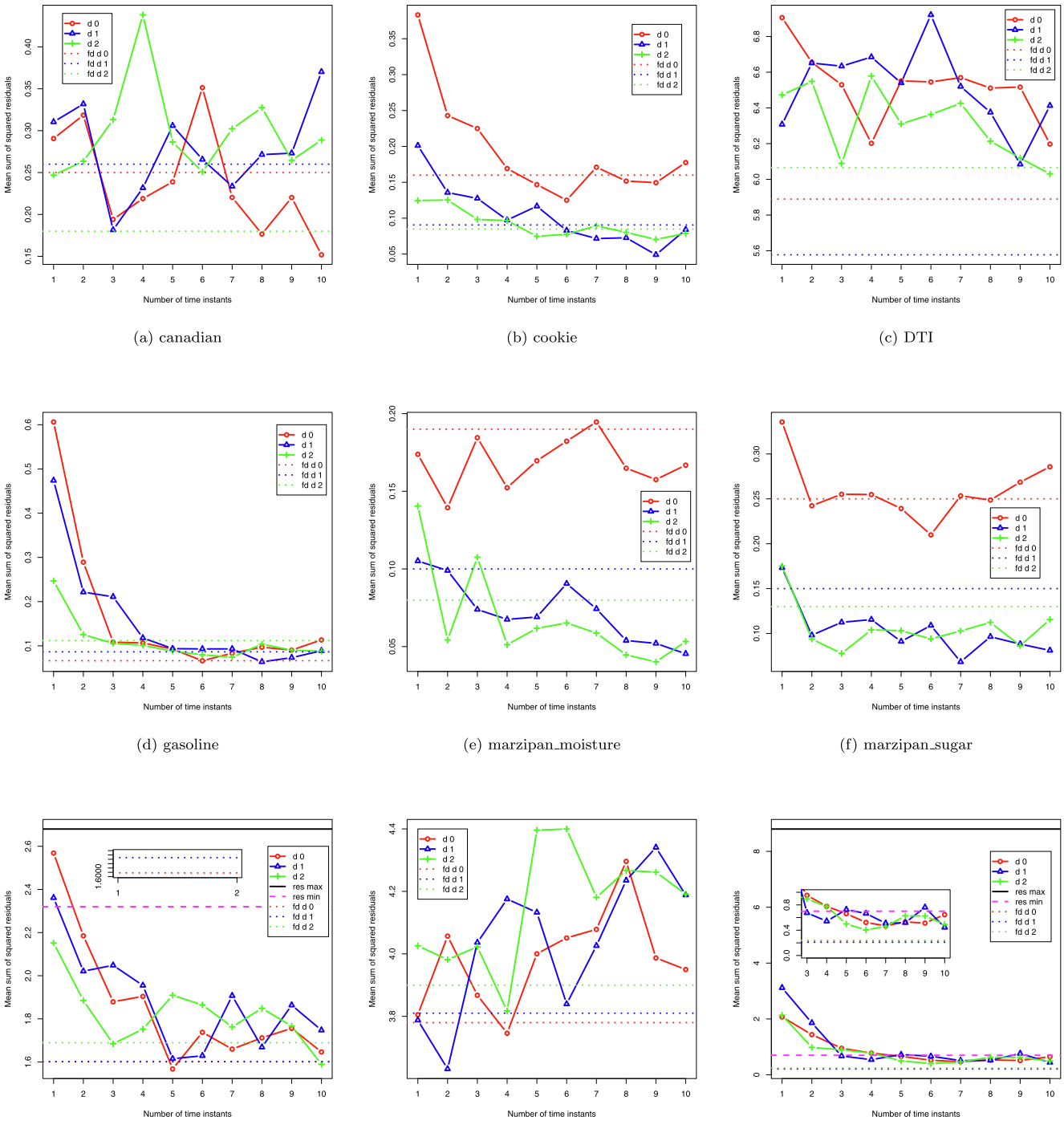
For the real applications data sets, Fig. 3 shows six out of nine data sets where the interval selection results are consistently better than the results in the full domain. This is true for small values of h and d. Indeed, the effect of the overfitting clearly appears for large values of h. It is worth noting that, in theory, the estimate of the derivatives given by Eq. (3) is accurate only if the monitoring is dense (i.e., the difference between t_h and t_{h-1} is sufficiently small) and the trajectories X are not too noisy. In the data sets where the sample trajectories are very rugged, the numerical estimates of the derivative may be unreliable, which may affect the stability of the results. This is the case of the *canadian* data set, for example. Nevertheless, observe that, even with such an irregular behavior in the results path, the quality of the regression results still improves the full domain residuals when compared with the interval selection results for $d = 1$. In any case, if the prediction results were believed to be affected, then other techniques which remove noise, such as De Brabanter et al. (2013) and the references therein should be considered.

In general, these results are very encouraging and show the potential of interval selection in a nonlinear setting, combined if needed with the use of the derivatives of the functional data.

For the case of instant selection ($\lambda = +\infty$), the results are more mixed. In simulated data, Fig. 4 shows three cases (*MK005, MK01* and *PSVone*) where instant selection provides better results than using the full domain, for a large enough value of H, for any choice of d. This behavior repeats in data set *FHV* for $d = 1, 2$, but not for $d = 0$.

5.3. Sensitivity analysis with respect to λ.

In this section, we will study how sensitive is the regularization parameter λ to the mean sum of the squared errors and to the length of intervals. To do this, we have run Algorithm 2 for $\lambda \in \left\{10^{-2}, \ldots, 10^{2}\right\}$. Fig. 9 shows than in three out five of the simulated data sets, better predictions in terms of the residuals are obtained when our approach is applied with $d = 1, 2$ instead of



(a) canadian (b) cookie (c) DTI

(d) gasoline (e) marzipan_moisture (f) marzipan_sugar

(g) sugar (h) sunflower (i) tecator

Fig. 5. Mean sum of squared errors for $\lambda = +\infty$, i.e., selecting time instants (intervals of zero length). Results on data from real applications.

using the information provided by the whole time interval (*full domain*). In particular, in the data set *FHV*, our method using the information until the second derivative ($d = 2$) is able to improve the results of the *full domain* counterpart for every value of h. Similar performances are obtained with the proposed approach with $d = 1$ in the data sets *MK005* and *MK01*.

In the case of the real-life applications, Fig. 10 shows that our proposal is able to get better predictions for at least one value of h in all the data sets (except *DTI*) when comparing with the *full domain* counterpart. For example, in the data set *marzipan_moisture*, the curve obtained with our approach for all the values of h and $d = 0$ is below the curve obtained with the information given by the whole time interval, indicating that the methodology presented in this paper yields better predictions. In contrast, we can state that, in data set *gasoline*, slightly better predictions are forecasted with our approach using $d = 0$ and $h = 1$, than

Table 2
Mean and standard deviation (in parentheses) of the sum of the squared errors on all data sets using $\lambda = 0$. The last column indicates the average value *H_opt* that would be selected according to the strategy proposed in Section 3.5.

FHV			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.09 (0.02)	0.08 (0.02)	0.08 (0.02)	0.08 (0.01)	0.1 (0.02)	0.14 (0.04)	0.15 (0.04)	0.16 (0.04)	0.18 (0.04)	0.19 (0.04)	0.21 (0.05)	2.1
			1	0.1 (0.03)	0.23 (0.26)	0.12 (0.08)	0.11 (0.03)	0.13 (0.03)	0.15 (0.04)	0.17 (0.06)	0.17 (0.06)	0.18 (0.06)	0.18 (0.06)	0.18 (0.06)	2.4
			2	0.23 (0.07)	0.12 (0.04)	0.13 (0.03)	0.12 (0.04)	0.14 (0.06)	0.15 (0.06)	0.16 (0.06)	0.17 (0.07)	0.19 (0.08)	0.21 (0.1)	0.25 (0.12)	1.6
MK005			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.55 (0.21)	0.71 (0.36)	0.66 (0.35)	0.61 (0.3)	0.68 (0.34)	0.72 (0.35)	0.72 (0.38)	0.73 (0.41)	0.74 (0.44)	0.74 (0.46)	0.74 (0.48)	3.5
			1	0.84 (0.48)	0.59 (0.18)	0.61 (0.26)	0.59 (0.29)	0.63 (0.31)	0.64 (0.32)	0.64 (0.31)	0.65 (0.32)	0.67 (0.33)	0.69 (0.34)	0.7 (0.36)	2
			2	0.82 (0.49)	0.58 (0.29)	0.57 (0.33)	0.65 (0.39)	0.63 (0.36)	0.63 (0.36)	0.63 (0.36)	0.63 (0.37)	0.63 (0.37)	0.63 (0.37)	0.64 (0.37)	1.5
MK01			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	1.34 (0.73)	1.48 (0.61)	1.48 (0.7)	1.51 (0.69)	1.48 (0.75)	1.57 (0.74)	1.58 (0.83)	1.67 (0.81)	1.68 (0.89)	1.71 (0.91)	1.73 (0.92)	2.7
			1	1.92 (1.19)	1.58 (0.99)	1.78 (1.54)	1.92 (1.74)	1.98 (1.77)	2.03 (1.76)	1.95 (1.68)	1.77 (1.13)	1.78 (1.14)	1.79 (1.14)	1.77 (1.15)	2.1
			2	1.8 (1.1)	1.55 (1.18)	1.52 (0.82)	1.83 (1.18)	1.91 (0.96)	1.94 (1.03)	1.97 (1.14)	2.11 (1.49)	2.12 (1.47)	2.12 (1.46)	2.11 (1.45)	1.1
PSVone			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.01 (0.01)	0.02 (0.02)	0.01 (0.02)	0.02 (0.03)	0.03 (0.07)	0.05 (0.11)	0.03 (0.04)	0.04 (0.07)	0.03 (0.03)	0.05 (0.05)	0.05 (0.05)	2.8
			1	0.01 (0.01)	0.02 (0.02)	0.04 (0.04)	0.03 (0.05)	0.02 (0.02)	0.04 (0.04)	0.05 (0.06)	0.04 (0.04)	0.05 (0.06)	0.08 (0.1)	0.11 (0.19)	2.4
			2	0.01 (0.01)	0.02 (0.02)	0.03 (0.04)	0.03 (0.02)	0.04 (0.05)	0.06 (0.07)	0.12 (0.17)	0.09 (0.11)	0.08 (0.11)	0.1 (0.11)	0.09 (0.11)	2
PSVthree			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.02 (0.02)	0.15 (0.29)	0.04 (0.02)	0.05 (0.05)	0.08 (0.12)	0.08 (0.1)	0.09 (0.13)	0.12 (0.2)	0.17 (0.26)	0.18 (0.27)	0.2 (0.29)	3.4
			1	0.05 (0.05)	0.17 (0.3)	0.15 (0.24)	0.16 (0.25)	0.19 (0.23)	0.2 (0.25)	0.25 (0.33)	0.32 (0.61)	0.25 (0.39)	0.33 (0.64)	0.26 (0.42)	2.2
			2	0.03 (0.04)	0.04 (0.04)	0.11 (0.18)	0.15 (0.23)	0.15 (0.22)	0.19 (0.27)	0.3 (0.42)	0.23 (0.34)	0.3 (0.48)	0.31 (0.48)	0.33 (0.48)	1.3
canadian	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0.31	0.47	0	0.25 (0.58)	0.38 (0.82)	0.22 (0.4)	0.3 (0.46)	0.21 (0.42)	0.22 (0.5)	0.27 (0.67)	0.22 (0.4)	0.19 (0.41)	0.24 (0.48)	0.23 (0.52)	4.4
			1	0.26 (0.55)	0.28 (0.55)	0.18 (0.28)	0.19 (0.36)	0.2 (0.38)	0.24 (0.47)	0.26 (0.53)	0.19 (0.36)	0.22 (0.44)	0.21 (0.37)	0.21 (0.32)	4.2
			2	0.18 (0.39)	0.19 (0.28)	0.32 (0.73)	0.33 (0.83)	0.23 (0.46)	0.2 (0.38)	0.25 (0.49)	0.21 (0.43)	0.2 (0.38)	0.21 (0.39)	0.19 (0.37)	3.6
cookie			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.16 (0.38)	0.12 (0.37)	0.13 (0.4)	0.16 (0.48)	0.19 (0.42)	0.19 (0.38)	0.23 (0.46)	0.23 (0.36)	0.26 (0.43)	0.28 (0.45)	0.25 (0.33)	2.1
			1	0.09 (0.29)	0.07 (0.26)	0.09 (0.36)	0.1 (0.46)	0.08 (0.17)	0.09 (0.28)	0.11 (0.27)	0.1 (0.26)	0.11 (0.27)	0.12 (0.29)	0.12 (0.29)	2.6
			2	0.08 (0.28)	0.09 (0.4)	0.07 (0.23)	0.1 (0.26)	0.09 (0.24)	0.11 (0.3)	0.1 (0.24)	0.1 (0.33)	0.12 (0.32)	0.13 (0.29)	0.14 (0.26)	2.5

DTI			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	5.88 (2.90)	6.35 (3.38)	6.42 (3.46)	5.98 (2.87)	6.06 (3.01)	6.33 (3.79)	5.96 (2.38)	6.02 (2.45)	6.15 (3.37)	5.96 (3.15)	6.4 (4.04)	2.4
			1	5.57 (3.03)	6.36 (3.75)	5.91 (2.62)	6.5 (3.57)	6.05 (3.26)	6.12 (3.52)	6.32 (3.82)	6.43 (3.74)	6.46 (3.69)	6.47 (3.69)	6.59 (4.08)	4.9
			2	6.06 (2.84)	6.3 (3.68)	6.25 (3.89)	6.61 (4.01)	6.63 (3.46)	6.21 (3.62)	6.44 (3.34)	6.39 (3.62)	6.17 (3.67)	6.32 (3.68)	6.28 (3.72)	3.2
gasoline			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.06 (0.08)	0.06 (0.1)	0.11 (0.22)	0.08 (0.1)	0.1 (0.17)	0.09 (0.13)	0.13 (0.18)	0.14 (0.2)	0.1 (0.17)	0.14 (0.19)	0.16 (0.2)	2.4
			1	0.08 (0.18)	0.07 (0.12)	0.09 (0.14)	0.11 (0.25)	0.1 (0.21)	0.14 (0.38)	0.17 (0.37)	0.18 (0.33)	0.16 (0.25)	0.17 (0.27)	0.36 (1.45)	2.5
			2	0.12 (0.2)	0.08 (0.15)	0.09 (0.17)	0.1 (0.15)	0.18 (0.49)	0.13 (0.31)	0.14 (0.21)	0.13 (0.41)	0.14 (0.21)	0.19 (0.39)	0.14 (0.23)	2.5
marzipan_moist.	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	4.8	11.8	0	0.19 (0.35)	0.11 (0.19)	0.14 (0.26)	0.21 (0.29)	0.23 (0.52)	0.24 (0.59)	0.26 (0.56)	0.23 (0.39)	0.34 (0.64)	0.3 (0.63)	0.28 (0.6)	2.7
			1	0.1 (0.25)	0.07 (0.13)	0.06 (0.12)	0.05 (0.11)	0.06 (0.13)	0.08 (0.16)	0.08 (0.17)	0.07 (0.15)	0.09 (0.11)	0.11 (0.16)	0.1 (0.15)	2.6
			2	0.08 (0.18)	0.06 (0.09)	0.08 (0.17)	0.07 (0.12)	0.1 (0.2)	0.14 (0.3)	0.1 (0.25)	0.07 (0.16)	0.09 (0.16)	0.1 (0.16)	0.09 (0.15)	3.2
marzipan_sugar	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0.34	0.58	0	0.25 (0.49)	0.25 (0.47)	0.38 (0.81)	0.24 (0.31)	0.33 (0.48)	0.33 (0.59)	0.3 (0.56)	0.39 (0.75)	0.42 (0.78)	0.34 (0.59)	0.29 (0.53)	2.5
			1	0.15 (0.23)	0.09 (0.19)	0.1 (0.16)	0.12 (0.16)	0.13 (0.21)	0.14 (0.22)	0.14 (0.22)	0.11 (0.17)	0.12 (0.19)	0.13 (0.19)	0.12 (0.2)	2.4
			2	0.13 (0.21)	0.11 (0.19)	0.12 (0.19)	0.1 (0.16)	0.1 (0.15)	0.13 (0.23)	0.13 (0.2)	0.17 (0.22)	0.14 (0.19)	0.14 (0.18)	0.14 (0.18)	3
sugar			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	1.6 (1.16)	1.8 (1.37)	1.83 (1.38)	1.84 (1.58)	1.84 (1.52)	1.82 (1.51)	1.83 (1.45)	1.86 (1.46)	1.84 (1.45)	1.85 (1.49)	1.87 (1.41)	2.3
			1	1.6 (1.42)	2.08 (1.64)	1.81 (1.56)	1.82 (1.47)	1.88 (1.6)	1.84 (1.63)	1.88 (1.65)	1.85 (1.58)	1.83 (1.49)	1.82 (1.48)	1.84 (1.53)	3.5
			2	1.68 (1.46)	1.76 (1.37)	1.88 (1.55)	1.92 (1.39)	1.9 (1.4)	1.9 (1.4)	1.9 (1.4)	1.88 (1.41)	1.86 (1.42)	1.86 (1.42)	1.88 (1.44)	4.8
sunflower			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	3.78 (3.62)	4.69 (4.31)	3.9 (3.4)	4.12 (4.32)	4.19 (4.33)	4.09 (3.91)	4.01 (3.59)	3.99 (3.51)	3.99 (3.51)	3.92 (3.49)	3.98 (3.5)	4.7
			1	3.81 (3.58)	4.29 (3.61)	3.86 (3.59)	4.11 (3.84)	4.32 (3.76)	4.24 (3.77)	4.08 (3.62)	4.07 (3.63)	4.01 (3.66)	3.95 (3.67)	3.98 (3.62)	4.1
			2	3.90 (3.57)	3.83 (3.47)	3.86 (3.65)	3.8 (3.63)	3.92 (3.67)	4.03 (3.63)	3.91 (3.73)	3.93 (3.7)	3.96 (3.63)	4.11 (3.62)	3.94 (3.57)	3
tecator	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	2.41	11.11	0	0.23 (0.14)	0.25 (0.14)	0.23 (0.17)	0.24 (0.2)	0.29 (0.32)	0.41 (0.49)	0.4 (0.42)	0.45 (0.46)	0.45 (0.42)	0.49 (0.43)	0.5 (0.42)	2.8
			1	0.21 (0.13)	0.47 (0.63)	0.24 (0.13)	0.37 (0.36)	0.38 (0.38)	0.42 (0.41)	0.47 (0.44)	0.49 (0.45)	0.5 (0.41)	0.52 (0.42)	0.53 (0.41)	2.6
			2	0.24 (0.17)	0.23 (0.13)	0.36 (0.27)	0.39 (0.29)	0.43 (0.33)	0.47 (0.36)	0.55 (0.37)	0.58 (0.39)	0.6 (0.4)	0.61 (0.38)	0.61 (0.37)	2

Table 3
Mean and standard deviation (in parentheses) of the sum of the squared errors on all datasets using $\lambda = +\infty$. The last column indicates the average value *H_opt* that would be selected according to the strategy proposed in Section 3.5.

FHV	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0.16 (0.05)	0.82 (0.18)	0	0.09 (0.02)	5.26 (1.48)	5.4 (1.19)	4.75 (1.29)	3.13 (0.71)	1.96 (0.7)	1.31 (0.53)	0.37 (0.17)	0.24 (0.14)	0.21 (0.15)	0.19 (0.07)	9.6
			1	0.1 (0.03)	6.58 (1.41)	3.84 (1.64)	1.84 (0.99)	0.18 (0.18)	0.1 (0.05)	0.09 (0.05)	0.09 (0.03)	0.09 (0.03)	0.09 (0.02)	0.09 (0.02)	7.8
			2	0.23 (0.07)	7.56 (1.61)	1.48 (0.27)	0.5 (0.17)	0.3 (0.11)	0.28 (0.14)	0.2 (0.05)	0.19 (0.05)	0.21 (0.05)	0.2 (0.06)	0.2 (0.06)	7.1
MK005			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.55 (0.21)	0.83 (0.32)	0.5 (0.23)	0.44 (0.18)	0.44 (0.2)	0.42 (0.19)	0.42 (0.16)	0.44 (0.18)	0.44 (0.2)	0.43 (0.18)	0.43 (0.17)	6.1
			1	0.84 (0.48)	0.83 (0.41)	0.46 (0.2)	0.43 (0.19)	0.45 (0.19)	0.44 (0.18)	0.44 (0.2)	0.42 (0.18)	0.43 (0.17)	0.42 (0.16)	0.42 (0.17)	6.9
			2	0.82 (0.49)	0.95 (0.44)	0.66 (0.39)	0.54 (0.3)	0.53 (0.33)	0.52 (0.28)	0.49 (0.23)	0.52 (0.24)	0.52 (0.23)	0.51 (0.24)	0.51 (0.22)	5.9
MK01			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	1.34 (0.73)	1.63 (0.85)	1.26 (0.73)	1.29 (0.87)	1.17 (0.66)	1.2 (0.71)	1.21 (0.74)	1.19 (0.66)	1.27 (0.85)	1.18 (0.63)	1.28 (0.74)	6.2
			1	1.92 (1.19)	1.53 (1.01)	1.21 (0.72)	1.27 (0.86)	1.27 (0.72)	1.26 (0.75)	1.26 (0.69)	1.35 (0.88)	1.29 (0.74)	1.29 (0.84)	1.3 (0.81)	3.1
			2	1.8 (1.1)	1.72 (0.96)	1.33 (0.76)	1.26 (0.76)	1.31 (0.74)	1.28 (0.81)	1.34 (0.81)	1.33 (0.76)	1.33 (0.78)	1.35 (0.83)	1.34 (0.82)	5.6
PSVone			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.01 (0.01)	0.27 (0.17)	0.07 (0.05)	0.05 (0.04)	0.05 (0.07)	0.02 (0.02)	0.01 (0.02)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	8.2
			1	0.01 (0.01)	0.29 (0.19)	0.12 (0.08)	0.05 (0.04)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	8.1
			2	0.01 (0.01)	0.25 (0.15)	0.14 (0.1)	0.06 (0.05)	0.03 (0.03)	0.02 (0.02)	0.01 (0.03)	0.01 (0.01)	0.01 (0.01)	0.02 (0.02)	0.01 (0.02)	7.7
PSVthree			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.02 (0.02)	1.61 (1.43)	0.92 (0.87)	0.29 (0.31)	0.15 (0.15)	0.09 (0.03)	0.06 (0.03)	0.07 (0.05)	0.07 (0.04)	0.06 (0.05)	0.05 (0.03)	9.4
			1	0.05 (0.05)	1.37 (0.86)	0.95 (0.83)	0.41 (0.36)	0.13 (0.08)	0.11 (0.09)	0.09 (0.07)	0.1 (0.09)	0.08 (0.09)	0.05 (0.04)	0.05 (0.04)	8.8
			2	0.03 (0.04)	1.71 (1.3)	0.82 (0.66)	0.69 (0.78)	0.27 (0.4)	0.13 (0.15)	0.07 (0.05)	0.06 (0.04)	0.06 (0.05)	0.05 (0.04)	0.05 (0.06)	9
canadian			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.25 (0.58)	0.29 (0.41)	0.31 (0.67)	0.19 (0.33)	0.21 (0.57)	0.23 (0.43)	0.35 (0.69)	0.22 (0.46)	0.17 (0.44)	0.22 (0.55)	0.15 (0.29)	6.1
			1	0.26 (0.55)	0.31 (0.52)	0.33 (0.81)	0.18 (0.35)	0.23 (0.49)	0.3 (0.57)	0.26 (0.34)	0.23 (0.43)	0.27 (0.88)	0.27 (0.46)	0.37 (0.95)	4.7
			2	0.18 (0.39)	0.24 (0.45)	0.26 (0.5)	0.31 (0.71)	0.43 (1.06)	0.28 (0.57)	0.25 (0.67)	0.3 (0.7)	0.32 (0.87)	0.26 (0.5)	0.28 (0.58)	4.3
cookie			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.16 (0.38)	0.38 (0.42)	0.24 (0.33)	0.22 (0.31)	0.16 (0.34)	0.14 (0.33)	0.12 (0.22)	0.17 (0.37)	0.15 (0.29)	0.14 (0.27)	0.17 (0.38)	7
			1	0.09 (0.29)	0.2 (0.43)	0.13 (0.38)	0.12 (0.26)	0.09 (0.33)	0.11 (0.52)	0.08 (0.26)	0.07 (0.22)	0.07 (0.21)	0.04 (0.08)	0.08 (0.35)	6.8
			2	0.08 (0.28)	0.12 (0.22)	0.12 (0.36)	0.09 (0.24)	0.09 (0.26)	0.07 (0.2)	0.07 (0.29)	0.08 (0.25)	0.08 (0.21)	0.07 (0.17)	0.07 (0.25)	6.5

DTI			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	5.88 (2.9)	6.9 (4.03)	6.65 (3.68)	6.52 (3.51)	6.2 (3.34)	6.55 (4.03)	6.54 (3.95)	6.56 (3.8)	6.51 (3.36)	6.51 (3.32)	6.19 (3.47)	7.1
			1	5.57 (3.03)	6.3 (2.91)	6.65 (3.82)	6.63 (3.51)	6.68 (3.6)	6.53 (3.75)	6.92 (4.11)	6.51 (3.92)	6.37 (3.75)	6.08 (3.87)	6.41 (3.68)	6
			2	6.06 (2.84)	6.47 (3.7)	6.54 (4.13)	6.08 (2.87)	6.57 (2.9)	6.31 (3.36)	6.36 (3.37)	6.42 (3.58)	6.21 (3.69)	6.11 (3.61)	6.03 (3.47)	5.9
gasoline			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.06 (0.08)	0.6 (1.3)	0.28 (0.57)	0.1 (0.13)	0.1 (0.25)	0.09 (0.14)	0.06 (0.09)	0.08 (0.19)	0.09 (0.2)	0.09 (0.26)	0.11 (0.3)	8
			1	0.08 (0.18)	0.47 (1.76)	0.22 (0.52)	0.21 (0.67)	0.11 (0.23)	0.09 (0.12)	0.09 (0.12)	0.09 (0.15)	0.06 (0.12)	0.07 (0.11)	0.08 (0.12)	7.3
			2	0.12 (0.2)	0.24 (0.42)	0.12 (0.16)	0.1 (0.14)	0.1 (0.16)	0.08 (0.17)	0.07 (0.16)	0.07 (0.14)	0.1 (0.2)	0.09 (0.15)	0.08 (0.15)	6.7
marzipan_moist.			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.19 (0.35)	0.17 (0.29)	0.13 (0.2)	0.18 (0.4)	0.15 (0.27)	0.16 (0.26)	0.18 (0.29)	0.19 (0.3)	0.16 (0.28)	0.15 (0.29)	0.16 (0.31)	5.7
			1	0.1 (0.25)	0.1 (0.17)	0.09 (0.16)	0.07 (0.12)	0.06 (0.1)	0.06 (0.09)	0.09 (0.15)	0.07 (0.1)	0.05 (0.09)	0.05 (0.07)	0.04 (0.04)	6.3
			2	0.08 (0.18)	0.14 (0.25)	0.05 (0.09)	0.1 (0.22)	0.05 (0.08)	0.06 (0.09)	0.06 (0.07)	0.05 (0.08)	0.04 (0.07)	0.04 (0.03)	0.05 (0.1)	5.4
marzipan_sugar			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	0.25 (0.49)	0.33 (0.43)	0.24 (0.37)	0.25 (0.38)	0.25 (0.39)	0.23 (0.34)	0.2 (0.29)	0.25 (0.32)	0.24 (0.33)	0.26 (0.37)	0.28 (0.36)	5.7
			1	0.15 (0.23)	0.17 (0.23)	0.09 (0.14)	0.11 (0.19)	0.11 (0.27)	0.09 (0.15)	0.1 (0.18)	0.06 (0.12)	0.09 (0.21)	0.08 (0.13)	0.08 (0.13)	6.1
			2	0.13 (0.21)	0.17 (0.32)	0.09 (0.13)	0.07 (0.1)	0.1 (0.14)	0.1 (0.17)	0.09 (0.17)	0.1 (0.18)	0.11 (0.25)	0.08 (0.16)	0.11 (0.2)	6.1
sugar	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	2.32	2.68	0	1.6 (1.16)	2.56 (1.69)	2.18 (1.26)	1.87 (1.45)	1.9 (1.46)	1.56 (0.97)	1.73 (1.41)	1.65 (1.1)	1.71 (1.29)	1.75 (1.29)	1.64 (1.15)	6.9
			1	1.6 (1.42)	2.36 (1.73)	2.02 (1.5)	2.04 (1.59)	1.95 (1.6)	1.61 (1.15)	1.62 (1.05)	1.9 (1.48)	1.66 (1.21)	1.86 (1.54)	1.74 (1.35)	7.2
			2	1.68 (1.46)	2.15 (1.63)	1.88 (1.6)	1.68 (1.23)	1.75 (1.33)	1.9 (1.59)	1.86 (1.56)	1.76 (1.41)	1.84 (1.48)	1.76 (1.41)	1.58 (1.15)	6.8
sunflower			*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
			0	3.78 (3.62)	3.8 (3.36)	4.05 (3.49)	3.86 (3.44)	3.74 (3.16)	4.00 (3.77)	4.05 (3.74)	4.07 (3.61)	4.29 (4.26)	3.98 (3.59)	3.95 (3.65)	4.6
			1	3.81 (3.58)	3.78 (3.34)	3.63 (3.09)	4.03 (3.47)	4.17 (3.8)	4.13 (3.46)	3.83 (3.49)	4.02 (3.75)	4.23 (3.82)	4.34 (3.73)	4.18 (3.66)	5.5
			2	3.90 (3.57)	4.02 (4.05)	3.98 (3.49)	4.02 (3.72)	3.81 (3.83)	4.39 (4.72)	4.39 (3.75)	4.18 (3.54)	4.26 (3.66)	4.26 (3.49)	4.19 (3.87)	4.2
tecator	res min	res max	*d*	full domain	*h*										
					1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0.7	8.8	0	0.23 (0.14)	2.07 (1)	1.43 (1.01)	0.95 (0.57)	0.77 (0.7)	0.66 (0.42)	0.52 (0.34)	0.46 (0.26)	0.53 (0.47)	0.51 (0.54)	0.64 (0.83)	8.1
			1	0.21 (0.13)	3.11 (2.45)	1.85 (1.12)	0.67 (0.36)	0.54 (0.26)	0.72 (0.47)	0.66 (0.51)	0.51 (0.31)	0.51 (0.24)	0.76 (0.54)	0.44 (0.43)	8.9
			2	0.24 (0.17)	2.13 (1.46)	0.98 (0.62)	0.89 (0.83)	0.77 (0.71)	0.49 (0.29)	0.4 (0.24)	0.45 (0.26)	0.62 (0.59)	0.62 (0.55)	0.49 (0.43)	8.1

Table 4
Mean and standard deviation (in parentheses) of the sum of the squared errors on all datasets using a grid in the parameter λ. The last column indicates the average value *H_opt* that would be selected according to the strategy proposed in Section 3.5.

FHV	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.09 (0.02)	0.07 (0.01)	0.07 (0.02)	0.08 (0.01)	0.1 (0.02)	0.12 (0.02)	0.13 (0.03)	0.16 (0.04)	0.18 (0.05)	0.19 (0.05)	0.21 (0.05)	2.2
	1	0.1 (0.03)	0.14 (0.04)	0.09 (0.02)	0.09 (0.01)	0.1 (0.02)	0.12 (0.03)	0.13 (0.04)	0.14 (0.04)	0.14 (0.04)	0.15 (0.04)	0.15 (0.04)	2.4
	2	0.23 (0.07)	0.1 (0.02)	0.11 (0.03)	0.11 (0.03)	0.12 (0.03)	0.13 (0.03)	0.14 (0.03)	0.15 (0.04)	0.17 (0.04)	0.18 (0.07)	0.19 (0.06)	1.7
MK005	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.55 (0.21)	0.66 (0.32)	0.64 (0.3)	0.63 (0.31)	0.63 (0.3)	0.67 (0.33)	0.69 (0.38)	0.72 (0.42)	0.71 (0.44)	0.74 (0.47)	0.76 (0.5)	3.1
	1	0.84 (0.48)	0.67 (0.39)	0.6 (0.3)	0.61 (0.3)	0.64 (0.33)	0.67 (0.38)	0.7 (0.39)	0.76 (0.4)	0.75 (0.41)	0.75 (0.43)	0.76 (0.43)	3.2
	2	0.82 (0.49)	0.52 (0.24)	0.56 (0.28)	0.6 (0.28)	0.57 (0.27)	0.56 (0.27)	0.55 (0.27)	0.55 (0.27)	0.55 (0.28)	0.57 (0.28)	0.57 (0.29)	2
MK01	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	1.34 (0.73)	1.58 (0.97)	1.49 (0.98)	1.48 (0.98)	1.63 (1.06)	1.61 (1.08)	1.6 (1.09)	1.6 (1.16)	1.65 (1.2)	1.6 (1.09)	1.66 (1.17)	3.3
	1	1.92 (1.19)	1.68 (1.35)	1.59 (0.99)	1.56 (0.93)	1.56 (1.16)	1.6 (1.27)	1.68 (1.32)	1.75 (1.37)	1.78 (1.41)	1.85 (1.43)	1.84 (1.49)	1.8
	2	1.8 (1.1)	1.55 (0.9)	1.43 (1.04)	1.41 (0.88)	1.64 (1)	1.77 (1.01)	1.82 (1.08)	1.8 (1.11)	1.87 (1.15)	1.89 (1.18)	1.91 (1.21)	2.1
PSVone	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.01 (0.01)	0.01 (0.01)	0.03 (0.03)	0.02 (0.02)	0.02 (0.02)	0.02 (0.02)	0.02 (0.02)	0.01 (0.01)	0.02 (0.03)	0.01 (0.01)	0.01 (0.01)	2.8
	1	0.01 (0.01)	0.03 (0.04)	0.02 (0.02)	0.02 (0.02)	0.02 (0.02)	0.02 (0.02)	0.03 (0.03)	0.02 (0.01)	0.02 (0.02)	0.03 (0.02)	0.04 (0.03)	3.4
	2	0.01 (0.01)	0.04 (0.06)	0.03 (0.02)	0.02 (0.02)	0.03 (0.02)	0.03 (0.02)	0.03 (0.03)	0.04 (0.03)	0.05 (0.04)	0.05 (0.04)	0.06 (0.04)	2.6
PSVthree	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.02 (0.02)	0.07 (0.05)	0.05 (0.04)	0.09 (0.14)	0.08 (0.1)	0.08 (0.11)	0.11 (0.21)	0.05 (0.07)	0.04 (0.03)	0.05 (0.04)	0.06 (0.05)	3.3
	1	0.05 (0.05)	0.17 (0.2)	0.21 (0.33)	0.15 (0.26)	0.31 (0.73)	0.32 (0.73)	0.3 (0.71)	0.31 (0.71)	0.32 (0.68)	0.32 (0.64)	0.33 (0.6)	3.6
	2	0.03 (0.04)	0.06 (0.06)	0.11 (0.13)	0.07 (0.06)	0.05 (0.03)	0.09 (0.1)	0.14 (0.17)	0.19 (0.18)	0.22 (0.19)	0.22 (0.21)	0.19 (0.24)	2.9
canadian	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	*H_opt*
	0	0.25 (0.58)	0.38 (1.24)	0.29 (0.56)	0.21 (0.36)	0.22 (0.44)	0.25 (0.46)	0.23 (0.44)	0.21 (0.42)	0.21 (0.34)	0.25 (0.44)	0.18 (0.28)	4.2
	1	0.26 (0.55)	0.29 (0.51)	0.29 (0.76)	0.28 (0.67)	0.26 (0.58)	0.25 (0.47)	0.15 (0.28)	0.21 (0.4)	0.23 (0.41)	0.21 (0.44)	0.24 (0.45)	3.2
	2	0.18 (0.39)	0.33 (0.78)	0.17 (0.24)	0.19 (0.29)	0.23 (0.53)	0.27 (0.55)	0.19 (0.34)	0.27 (0.49)	0.2 (0.37)	0.24 (0.37)	0.2 (0.38)	3.5
cookie	d	full domain	h										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.16 (0.38)	0.13 (0.42)	0.1 (0.3)	0.15 (0.45)	0.16 (0.35)	0.16 (0.32)	0.17 (0.38)	0.2 (0.31)	0.23 (0.36)	0.21 (0.37)	0.23 (0.35)	2.1
	1	0.09 (0.29)	0.07 (0.26)	0.07 (0.25)	0.07 (0.22)	0.07 (0.24)	0.1 (0.29)	0.09 (0.26)	0.1 (0.25)	0.12 (0.27)	0.1 (0.28)	0.11 (0.28)	2.6
	2	0.08 (0.28)	0.09 (0.38)	0.09 (0.25)	0.06 (0.17)	0.09 (0.29)	0.07 (0.15)	0.09 (0.31)	0.1 (0.32)	0.1 (0.32)	0.11 (0.33)	0.1 (0.27)	2.5

DTI	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	5.88 (2.9)	6.05 (3.47)	6.15 (3.73)	6.17 (4.05)	6.17 (3.89)	6.37 (4.04)	6.41 (3.67)	6.4 (3.81)	6.48 (3.75)	6.32 (3.8)	6.4 (3.78)	5.5
	1	5.57 (3.03)	6.43 (4.00)	6.24 (4.04)	6.04 (3.39)	5.8 (3.65)	5.88 (3.13)	6.14 (3.7)	5.94 (3.12)	6.35 (3.96)	5.97 (3.1)	6.08 (2.96)	3.7
	2	6.06 (2.84)	6.68 (4.00)	6.27 (3.72)	6.36 (3.86)	6.22 (3.81)	6.36 (4.3)	6.42 (4.15)	6.34 (4.23)	6.54 (4.24)	6.43 (4.19)	6.53 (4.03)	3.9
gasoline	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.06 (0.08)	0.06 (0.1)	0.08 (0.13)	0.08 (0.12)	0.08 (0.11)	0.09 (0.13)	0.09 (0.12)	0.08 (0.12)	0.09 (0.11)	0.11 (0.18)	0.13 (0.17)	2.5
	1	0.08 (0.18)	0.07 (0.1)	0.05 (0.08)	0.07 (0.1)	0.08 (0.1)	0.08 (0.15)	0.08 (0.12)	0.08 (0.15)	0.09 (0.11)	0.08 (0.17)	0.09 (0.14)	2.4
	2	0.12 (0.2)	0.09 (0.17)	0.09 (0.13)	0.07 (0.12)	0.08 (0.15)	0.06 (0.15)	0.07 (0.14)	0.06 (0.18)	0.08 (0.19)	0.1 (0.17)	0.09 (0.15)	2.2
marzipan_moist.	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.19 (0.35)	0.13 (0.23)	0.14 (0.24)	0.16 (0.3)	0.16 (0.24)	0.15 (0.28)	0.14 (0.26)	0.16 (0.29)	0.15 (0.27)	0.15 (0.26)	0.16 (0.27)	2.3
	1	0.1 (0.25)	0.06 (0.13)	0.06 (0.13)	0.09 (0.19)	0.1 (0.24)	0.11 (0.26)	0.09 (0.26)	0.08 (0.15)	0.1 (0.13)	0.12 (0.14)	0.11 (0.12)	2.9
	2	0.08 (0.18)	0.08 (0.13)	0.04 (0.06)	0.04 (0.05)	0.08 (0.15)	0.09 (0.14)	0.09 (0.15)	0.05 (0.07)	0.1 (0.15)	0.09 (0.14)	0.09 (0.15)	2.9
marzipan_sugar	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.25 (0.49)	0.24 (0.56)	0.15 (0.31)	0.15 (0.23)	0.26 (0.39)	0.22 (0.37)	0.31 (0.44)	0.33 (0.54)	0.31 (0.55)	0.35 (0.53)	0.3 (0.43)	3.1
	1	0.15 (0.23)	0.1 (0.16)	0.12 (0.17)	0.11 (0.17)	0.14 (0.23)	0.12 (0.17)	0.16 (0.24)	0.16 (0.23)	0.14 (0.22)	0.15 (0.22)	0.13 (0.2)	2.5
	2	0.13 (0.21)	0.11 (0.16)	0.11 (0.19)	0.18 (0.29)	0.12 (0.22)	0.11 (0.17)	0.15 (0.26)	0.14 (0.17)	0.17 (0.18)	0.13 (0.22)	0.13 (0.16)	2.2
sugar	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	1.6 (1.16)	1.69 (1.23)	1.61 (1.25)	1.63 (1.36)	1.51 (1.19)	1.55 (1.13)	1.53 (1.12)	1.61 (1.24)	1.66 (1.33)	1.58 (1.18)	1.61 (1.19)	4.1
	1	1.6 (1.42)	1.84 (1.54)	1.58 (1.19)	1.78 (1.43)	1.66 (1.2)	1.69 (1.33)	1.82 (1.39)	1.67 (1.24)	1.6 (1.17)	1.63 (1.17)	1.65 (1.17)	2.5
	2	1.68 (1.46)	1.81 (1.42)	1.79 (1.27)	1.99 (1.75)	1.9 (1.49)	1.9 (1.63)	1.81 (1.47)	1.72 (1.32)	1.72 (1.29)	1.79 (1.27)	1.84 (1.3)	4.1
sunflower	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	3.78 (3.62)	4.34 (4.02)	4.13 (4.27)	4.25 (3.94)	4.26 (3.9)	4.39 (4.57)	4.33 (4.75)	4.26 (4.75)	4.23 (4.71)	4.19 (4.67)	4.35 (4.61)	4.4
	1	3.81 (3.58)	4.13 (4.08)	4.7 (4.22)	3.96 (3.6)	4.13 (3.43)	4.05 (4.11)	3.79 (3.09)	3.73 (3.1)	3.7 (3.12)	3.8 (3.27)	3.95 (3.54)	3.4
	2	3.90 (3.57)	4.57 (3.64)	3.95 (3.48)	4.12 (3.64)	4.09 (3.99)	3.87 (3.33)	4.09 (3.42)	3.95 (3.52)	4.3 (3.8)	4.24 (3.72)	4.05 (3.57)	4.2
tecator	*d*	full domain	*h*										
			1	2	3	4	5	6	7	8	9	10	Av. *H_opt*
	0	0.23 (0.14)	0.31 (0.26)	0.2 (0.14)	0.21 (0.12)	0.25 (0.14)	0.27 (0.2)	0.34 (0.33)	0.35 (0.34)	0.35 (0.35)	0.39 (0.36)	0.39 (0.36)	2.6
	1	0.21 (0.13)	0.3 (0.15)	0.31 (0.33)	0.3 (0.35)	0.31 (0.33)	0.33 (0.25)	0.4 (0.45)	0.51 (0.46)	0.49 (0.42)	0.48 (0.42)	0.48 (0.42)	3.1
	2	0.24 (0.17)	0.25 (0.17)	0.28 (0.29)	0.22 (0.08)	0.31 (0.23)	0.45 (0.51)	0.36 (0.2)	0.58 (0.51)	0.52 (0.42)	0.5 (0.39)	0.49 (0.33)	3.4

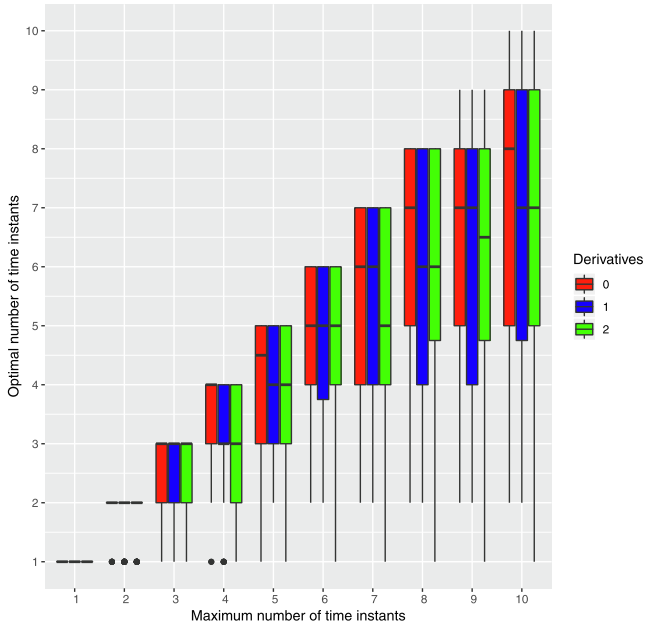


Fig. 6. Boxplots of the *H_opt* value obtained in data set *cookie* running Problem (9) from $h = 1$ to $h = 10$ and $\lambda = +\infty$.

those obtained when the *full domain* method is applied. Regarding the results given by our approach for $d = 1$, we can observe that in some data sets, in order to get better forecasting results than thoses given by the information provided by the *full domain*, we have to pay attention to lower values of h, e.g., *cookie*, *gasoline*, *marzipan_moisture*, and *marzipan_sugar*. Nevertheless, in the data set *sunfower*, we have to restrict to $h \geqslant 6$ in our approach if better results are desired. Finally, the proposed method for $d = 2$ is better than the *full domain* counterpart in data sets *cookie* and *tecator* with $h = 3$, for instance. In the data set *gasoline*, the curve provided by our approach give better results than the one given by the whole time domain for any value of h.

Hence, in general, we can state that including a penalization of the interval length in the optimization problem, is a good way of improving the forecasting.

In addition, Fig. 8(a) and (b) show the average over all the folds of the mean sum of the squared errors estimated on sample s_3 and s_4, respectively when Algorithm 2 is run for $\varepsilon = 10^{-3}$, the information until the first derivative is used, i.e., $d = 1$, and the number of intervals sought is $h = 1$ in the database *marzipan_sugar*. We observe that the plot is approximately U-shaped, which means that the best values of the parameter are thoses in the middle of the λ grid. That is, penalizing the length of the intervals, with a penalizing trade-off parameter λ, has the potential of improving the results shown in Section 5, providing better predictions than both selecting only time instants (intervals of length zero) and selecting intervals of unpenalized length. Moreover, we see in Fig. 8(c) the average length of the intervals over all the folds using the same settings as before, i.e., $\varepsilon = 10^{-3}$, $d = 1$ and $h = 1$. As expected, it can be seen that the larger the value of λ, then the shorter the interval length.

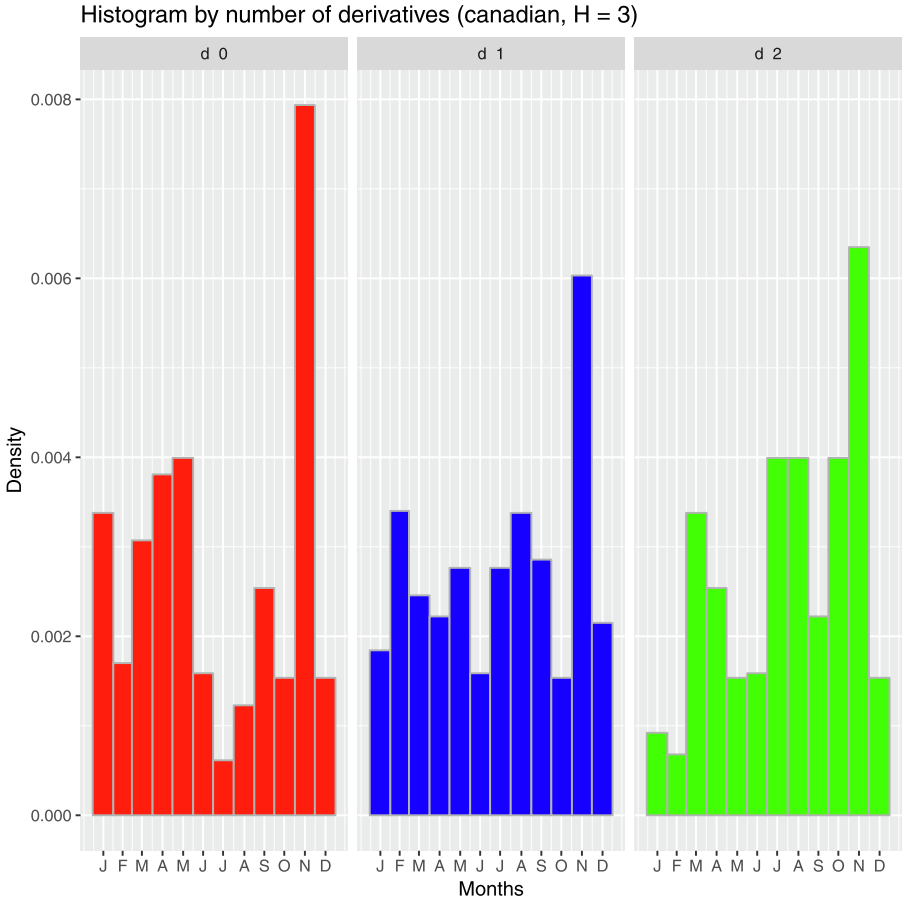
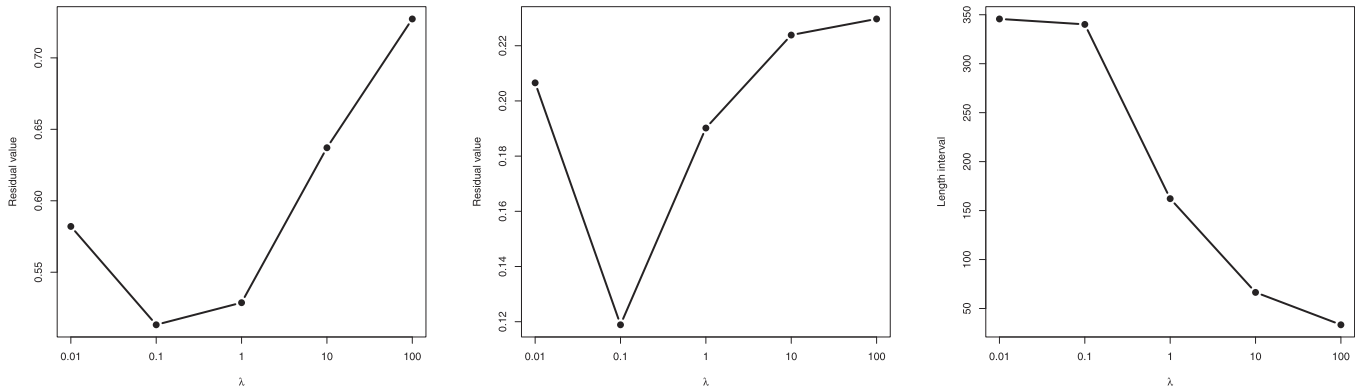
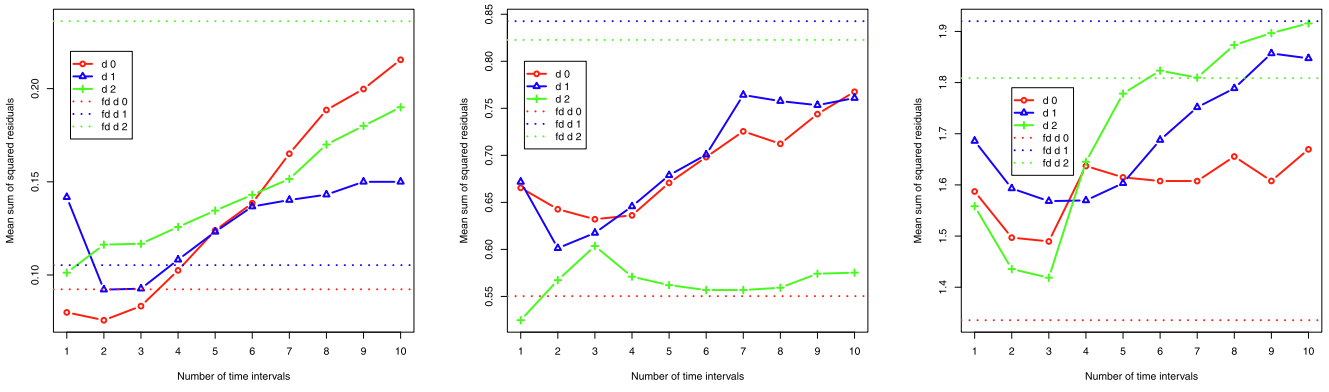


Fig. 7. Histogram of the time instants values in *canadian* data set when $h = 3$ time instants are sought ($\lambda = +\infty$).

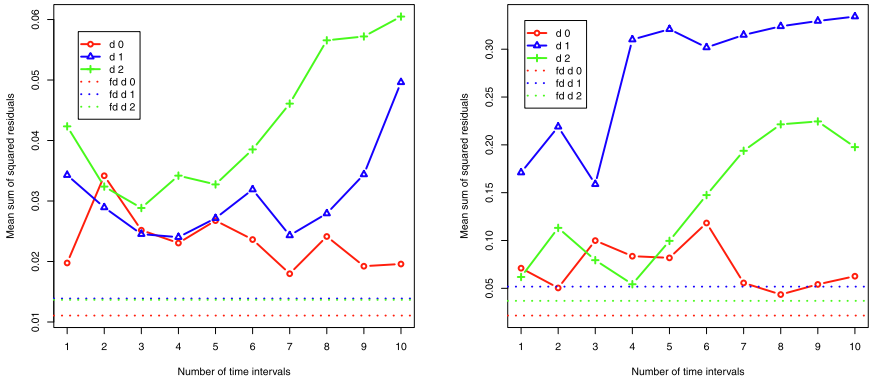


(a) mean squared residual error validation, s_3 (b) mean squared residual error testing, s_4 (c) mean length intervals

Fig. 8. Mean squared residual error on sample *s_3*, and *s_4* and mean length intervals in terms of the λ when our methodology is run in the data set *marzipan_sugar* for $\varepsilon = 10^{-3}$, $d = 1$ and $h = 1$.



(a) FHV (b) MK005 (c) MK01



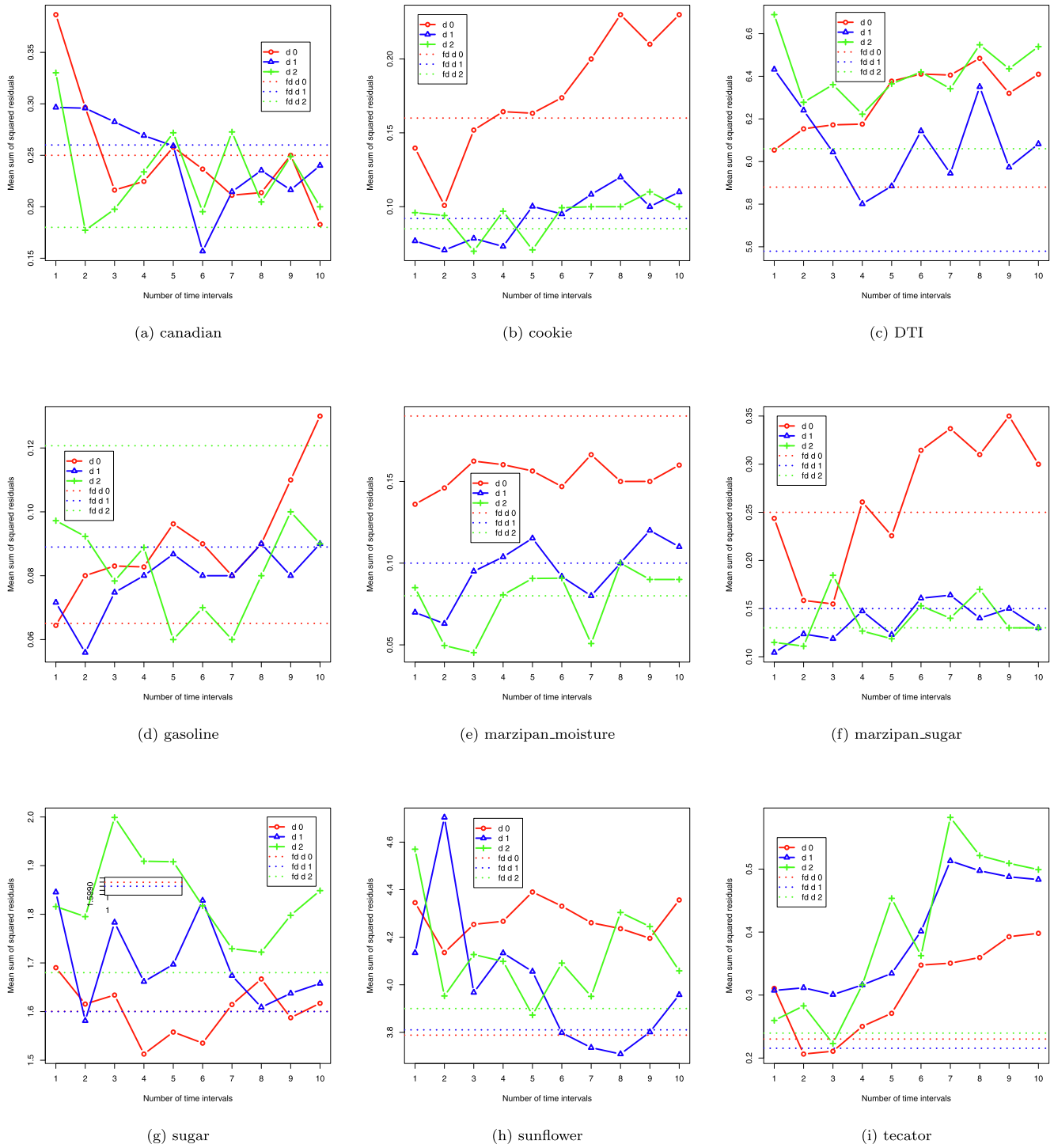
(d) PSVone (e) PSVthree

Fig. 9. Mean sum of squared errors for λ in a grid. Results on simulated data.

6. Conclusions and extensions

In this paper, we have proposed an approach based on continuous optimization to select time intervals in regression problems with (multivariate) functional data. Particularly, we formulate an optimization problem whose objective function has a regularization term that penalizes large intervals by means of a parameter that should be properly chosen. Hence, our methodology gives flexibility to the state-of-the-art models, since using exactly the same model, we can obtain the most relevant time instants, i.e., zero-length intervals when the regularization parameter tends to infinity, and also intervals of any length when the regularization parameter is equal to zero.

Furthermore, our proposal allows, in the very same manner, to add high-order information provided by the derivatives of the (multivariate) functional data. Such information is crucial, as has been shown in the numerical experience.



(a) canadian (b) cookie (c) DTI

(d) gasoline (e) marzipan_moisture (f) marzipan_sugar

(g) sugar (h) sunflower (i) tecator

Fig. 10. Mean sum of squared errors for λ in a grid. Results on real-life applications.

Indeed, our experiments show that our methodology outperforms existing results in the literature for a large enough h. When compared with using the whole domain, there is a clear benefit from interval selection in terms of predictive ability. In all, but one case analyzed, there is a wide range of values of h for which interval selection outperforms the use of the full domain and results in the literature. This suggests that the choice of H_opt is not, in general, a crucial point. In the only data set in which the selection of H_opt seems crucial, this only happens for $d = 0, 1$, suggesting that the use of higher order derivatives may overcome the issues with the selection of H_opt. Finally, we have shown that a regularization term based on interval length can yield even better results.

Nevertheless, our approach presents some flaws which deserve further research, such as a lack of stability of the hyperparameter λ, and the large deviations of results for some data sets. Hence, some

extensions of the present work are possible, such as a robust version of our proposal, which produces more stable results independent of the choice of the hyperparameters. Such extensions are far from trivial and require research that is beyond the scope of this manuscript. Hence, some extensions of the present work are possible, such as a more robust version of our proposal, which produce stable results independent for the choice of the hyperparameters. Finally, here we have just considered pure (multivariate) functional data. Our proposal can be easily extended to the hybrid multivariate case with a simple modification of the kernel function, (Jiménez-Cordero and Maldonado, 2020).

A more challenging topic seems to be the extension of our approach to spatio-temporal data, in which one seeks the most relevant time intervals and locations, or to other Data Science problems, such as clustering.

CRediT authorship contribution statement

Rafael Blanquero: Conceptualization, Methodology, Investigation, Writing - review & editing, Supervision, Funding acquisition. **Emilio Carrizosa:** Conceptualization, Methodology, Investigation, Writing - review & editing, Supervision, Funding acquisition. **Asunción Jiménez-Cordero:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Belén Martín-Barragán:** Conceptualization, Methodology, Investigation, Writing - review & editing, Supervision, Funding acquisition.

Acknowledgments

Research partially supported by research grants MTM2015-65915-R (Ministerio de Ciencia e Innovación, Spain), P11-FQM-7603, P18-FR-2369, FQM329 (Junta de Andalucía, Spain), FPU (Ministerio de Educación, Cultura y Deporte), all with EU ERDF funds, as well as FBBVA-COSECLA, and EP/R00370X/1 (Engineering and Physical Science Research Council, United Kingdom). This support is gratefully acknowledged. We also thank the team of the Scientific Computing Center of Andalucía (CICA) for the computing services provided.

References

Aguilera, A.M., Ocaña, F.A., Valderrama, M.J., 1997. An approximated principal component prediction model for continuous-time stochastic processes. Applied Stochastic Models and Data Analysis 13, 61–72.
Andersen, C.M., Bro, R., 2010. Variable selection in regression–a tutorial. Journal of Chemometrics 24, 728–737.
Aneiros, G., Vieu, P., 2014. Variable selection in infinite-dimensional problems. Statistics & Probability Letters 94, 12–20.
Aneiros, G., Vieu, P., 2016. Sparse nonparametric model for regression with functional covariate. Journal of Nonparametric Statistics 28, 839–859.
Aytug, H., 2015. Feature selection for support vector machines using generalized Benders decomposition. European Journal of Operational Research 244, 210–218.
Benítez-Peña, S., Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., 2019. Cost-sensitive feature selection for support vector machines. Computers & Operations Research 106, 169–178.
Berrendero, J.R., Bueno-Larraz, B., Cuevas, A., 2019. An RKHS model for variable selection in functional linear regression. Journal of Multivariate Analysis 170, 25–45.
Bertolazzi, P., Felici, G., Festa, P., Fiscon, G., Weitschek, E., 2016. Integer programming models for feature selection: New extensions and a randomized solution algorithm. European Journal of Operational Research 250, 389–399.
Blanquero, R., Carrizosa, E., Chis, O., Esteban, N., Jiménez-Cordero, A., Rodríguez, J.F., Sillero-Denamiel, M.R., 2016a. On extreme concentrations in chemical reaction networks with incomplete measurements. Industrial & Engineering Chemistry Research 55, 11417–11430.
Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., Rodríguez, J.F., 2016b. A global optimization method for model selection in chemical reactions networks. Computers & Chemical Engineering 93, 52–62.
Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., Martín-Barragán, B., 2019a. Functional-bandwidth kernel for Support Vector Machine with functional data: an alternating optimization algorithm. European Journal of Operational Research 275, 195–207.
Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., Martín-Barragán, B., 2019b. Variable selection in classification for multivariate functional data. Information Sciences 481, 445–462.
Carrizosa, E., Romero Morales, D., 2013. Supervised classification and mathematical optimization. Computers & Operations Research 40, 150–165.
Carrizosa, E., Martín-Barragán, B., Romero Morales, D., 2011. Detecting relevant variables and interactions in supervised classification. European Journal of Operational Research 213, 260–269.
Carrizosa, E., Martín-Barragán, B., Romero Morales, D., 2014. A nested heuristic for parameter tuning in support vector machines. Computers & Operations Research 43, 328–334.
Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. Computers & Electrical Engineering 40, 16–28.
Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., 2002. Choosing multiple parameters for support vector machines. Machine Learning 46, 131–159.
Colson, B., Marcotte, P., Savard, G., 2007. An overview of bilevel optimization. Annals of Operations Research 153, 235–256.
Comenetz, M., 2002. Calculus. The Elements. World Scientific. https://doi.org/10.1142/4920.
Core Team, R. 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna, Austria. URL: https://www.R-project.org/..
Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.
Dash, M., Liu, H., 1997. Feature selection for classification. Intelligent Data Analysis 1, 131–156.
Dash, M., Liu, H., 2000. Feature selection for clustering. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 110–121.
Dash, M., Liu, H., Yao, J., 1997. Dimensionality reduction of unsupervised data. In: Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp. 532–539.
Dash, M., Choi, K., Scheuermann, P., Liu, H., 2002. Feature selection for clustering-a filter solution. In: Proceedings. 2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. IEEE, pp. 115–122.
De Boor, C., 1978. A Practical Guide to Splines Volume 27 of Applied Mathematical Sciences. Springer-Verlag, New York.
De Brabanter, K., De Brabanter, J., De Moor, B., Gijbels, I., 2013. Derivative estimation with local polynomial fitting. The Journal of Machine Learning Research 14, 281–301.
Delaigle, A., Hall, P., 2012. Methodology and theory for partial least squares applied to functional data. The Annals of Statistics 40, 322–352.
Ferraty, F., Vieu, P., 2004. Nonparametric models for functional data, with application in regression, time series prediction and curve discrimination. Nonparametric Statistics 16, 111–125.
Ferraty, F., Vieu, P., 2006. Nonparametric Functional Data Analysis: Theory and Practice. Springer Science & Business Media.
Ferraty, F., Hall, P., Vieu, P., 2010. Most-predictive design points for functional data predictors. Biometrika 97, 807–824.
Fraiman, R., Gimenez, Y., Svarc, M. 2016. Feature selection for functional data. Journal of Multivariate Analysis, 146, 191–208. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces..
Friedman, J., Hastie, T., Tibshirani, R., 2001. The Elements of Statistical Learning Volume 1 of Springer Series in Statistics. Springer, Berlin.
Friedrichs, F., Igel, C. 2005. Evolutionary tuning of multiple svm parameters. Neurocomputing, 64, 107–117. Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004..
Goldsmith, J., Scheipl, F., 2014. Estimator selection and combination in scalar-on-function regression. Computational Statistics & Data Analysis 70, 362–372.
Grollemund, P.-M., Abraham, C., Baragatti, M., Pudlo, P., 2019. Bayesian functional linear regression with sparse step functions. Bayesian Analysis 14, 111–135.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182.
Hernández, N., Biscay, R.J., Talavera, I. 2007. Support vector regression methods for functional data. In: Rueda, L., Mery, D., Kittler, J. (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. CIARP 2007. Lecture Notes in Computer Science, Berlin Heidelberg volume 4756, Springer, pp. 564–573..
Hernández, N., Talavera, I., Biscay, R.J., Porro, D., Ferreira, M.M., 2009. Support vector regression for functional data in multivariate calibration problems. Analytica Chimica Acta 642, 110–116.
Hofmann, T., Schölkopf, B., Smola, A.J., 2008. Kernel methods in machine learning. The Annals of Statistics 36, 1171–1220.
Hsu, H.-H., Hsieh, C.-W., Lu, M.-D., 2011. Hybrid feature selection by combining filters and wrappers. Expert Systems with Applications 38, 8144–8150.
Hua, J., Tembe, W.D., Dougherty, E.R., 2009. Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition 42, 409–424.
James, G.M., Wang, J., Zhu, J., 2009. Functional linear regression that's interpretable. The Annals of Statistics 37, 2083–2108.
Jiménez-Cordero, A., Maldonado, S., 2020. Automatic Feature Scaling and Selection for Support Vector Machine Classification with Functional Data. Applied Intelligence. https://doi.org/10.1007/s10489-020-01765-6.
Johnstone, I.M., Lu, A.Y., 2009. On consistency and sparsity for principal components analysis in high dimensions. Journal of the American Statistical Association 104, 682–693.

Kaneko, H., Funatsu, K., 2015. Fast optimization of hyperparameters for support vector regression models with highly predictive ability. Chemometrics and Intelligent Laboratory Systems 142, 64–69.
Karagiannopoulos, M., Anyfantis, D., Kotsiantis, S., Pintelas, P. 2007. Feature selection for regression problems. In: Proceedings of the 8th Hellenic European Research on Computer Mathematics & its Applications, Athens, Greece, vol. 2022..
Keerthi, S.S., Lin, C.-J., 2003. Asymptotic behaviors of support vector machines with gaussian kernel. Neural Computation 15, 1667–1689.
Kneip, A., Poß, D., Sarda, P., et al., 2016. Functional linear regression with points of impact. The Annals of Statistics 44, 1–30.
Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97, 273–324.
Law, M.H., Jain, A.K., Figueiredo, M. 2003. Feature selection in mixture-based clustering. In: Advances in Neural Information Processing Systems, pp. 641–648..
Law, M.H., Figueiredo, M.A., Jain, A.K., 2004. Simultaneous feature selection and clustering using mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1154–1166.
Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., Nowe, A., 2012. A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 9, 1106–1119.
Li, Y., Dong, M., Hua, J., 2008. Localized feature selection for clustering. Pattern Recognition Letters 29, 10–18.
Li, G.-Z., Meng, H.-H., Yang, M.Q., Yang, J.Y., 2009. Combining support vector regression with feature selection for multivariate calibration. Neural Computing and Applications 18, 813–820.
Maldonado, S., Weber, R., 2009. A wrapper method for feature selection using support vector machines. Information Sciences 179, 2208–2217.
Maldonado, S., Weber, R., 2010. Feature selection for support vector regression via kernel penalization. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–7.
Maldonado, S., Weber, R., Basak, J., 2011. Simultaneous feature selection and classification using kernel-penalized support vector machines. Information Sciences 181, 115–128.
Martín-Barragán, B., Lillo, R., Romo, J., 2014. Interpretable support vector machines for functional data. European Journal of Operational Research 232, 146–155.
Matsui, H., Konishi, S., 2011. Variable selection for functional regression models via the L_1) regularization. Computational Statistics & Data Analysis 55, 3304–3310.
McKeague, I.W., Sen, B., 2010. Fractals with point impact in functional linear regression. Annals of Statistics 38, 2559–2586.
Mehmood, T., Liland, K.H., Snipen, L., Sæbø, S., 2012. A review of variable selection methods in partial least squares regression. Chemometrics and Intelligent Laboratory Systems 118, 62–69.
Molina, L.C., Belanche, L., Nebot, À., 2002. Feature selection algorithms: A survey and experimental evaluation. In: IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings. 2002. IEEE, pp. 306–313.
Müller, H.-G., Stadtmüller, U., 2005. Generalized functional linear models. Annals of Statistics 33, 774–805.
Muñoz, A., González, J., 2010. Representing functional data using support vector machines. Pattern Recognition Letters 31, 511–516.
Park, A.Y., Aston, J.A., Ferraty, F. 2016. Stable and predictive functional domain selection with application to brain images. arXiv preprint arXiv:1606.02186, URL:https://arxiv.org/abs/1606.02186. arXiv:1606.02186..
Phienthrakul, T., Kijsirikul, B. 2005. Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation GECCO '05,ACM, pp. 905–911..
Picheny, V., Servien, R., Villa-Vialaneix, N., 2019. Interpretable sparse SIR for functional data. Statistics and Computing 29, 255–267.
Preda, C., Saporta, G., 2005. Clusterwise PLS regression on a stochastic process. Computational Statistics & Data Analysis 49, 99–108.
Preda, C., Saporta, G., 2005. PLS regression on a stochastic process. Computational Statistics & Data Analysis 48, 149–158.
Ramsay, J.O., Dalzell, C.J., 1991. Some tools for functional data analysis. Journal of the Royal Statistical Society. Series B (Methodological) 53, 539–572.
Ramsay, J.O., Silverman, B.W., 2002. Applied Functional Data Analysis: Methods and Case Studies Volume 77 of Springer Series in Statistics. Springer-Verlag.
Ramsay, J.O., Silverman, B.W., 2005. Functional Data Analysis. Springer Series in Statistics. Springer-Verlag.
Reiss, P.T., Ogden, R.T., 2007. Functional principal component regression and functional partial least squares. Journal of the American Statistical Association 102, 984–996.
Rossi, F., Conan-Guez, B., 2005. Functional multi-layer perceptron: a non-linear tool for functional data analysis. Neural Networks 18, 45–60.
Rossi, F., Villa, N., 2006. Support vector machine for functional data classification. Neurocomputing 69, 730–742.
Schwarz, G., 1978. Estimating the Dimension of a Model. The Annals of Statistics 6, 461–464.
Smets, K., Verdonk, B., Jordaan, E.M., 2007. Evaluation of performance measures for SVR hyperparameter selection. In: 2007 International Joint Conference on Neural Networks, pp. 637–642.
Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14, 199–222.
Sreekumar, S., Verma, J., A, S., Kumar, R. 2016. Comparative analysis of intelligently tuned support vector regression models for short term load forecasting in smart grid framework. Technology and Economics of Smart Grids and Sustainable Energy, 2, 1..
Torrecilla Noguerales, J.L. 2015. On the theory and practice of variable selection for functional data. Ph.D. thesis. Universidad Autónoma de Madrid..
Tutz, G., Gertheiss, J., 2010. Feature extraction in signal regression: A boosting technique for functional data regression. Journal of Computational and Graphical Statistics 19, 154–174.
Van Dijck, G., Van Hulle, M.M., 2006. Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. In: International Conference on Artificial Neural Networks. Springer, pp. 31–40.
Yang, J.-B., Ong, C.-J., 2011. Feature selection using probabilistic prediction of support vector regression. IEEE Transactions on Neural Networks 22, 954–962.
Zhang, T., 2009. On the consistency of feature selection using greedy least squares regression. Journal of Machine Learning Research 10, 555–568.