Information Sciences 481 (2019) 445–462

Contents lists available at ScienceDirect

Information Sciences

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

# Variable selection in classification for multivariate functional data

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

[a] Facultad de Matemáticas, Departamento de Estadística e Investigación Operativa and Instituto de Matemáticas de la Universidad de Sevilla (IMUS) C/ Tarfia s/n Seville 41012, Spain
[b] University of Edinburgh Business School, 29 Buccleuch Place, Edinburgh EH89JS, UK

ARTICLE INFO

*Article history:*
Received 5 December 2017
Revised 25 December 2018
Accepted 26 December 2018
Available online 27 December 2018

*Keywords:*
Feature selection
Multivariate functional data analysis
Support Vector Machines

ABSTRACT

When classification methods are applied to high-dimensional data, selecting a subset of the predictors may lead to an improvement in the predictive ability of the estimated model, in addition to reducing the model complexity. In Functional Data Analysis (FDA), i.e., when data are functions, selecting a subset of predictors corresponds to selecting a subset of individual time instants in the time interval in which the functional data are measured. In this paper, we address the problem of selecting the most informative time instants in multivariate functional data, a case much less studied than its single-variate counterpart. Our proposal allows one to use in a very simple way high-order information of the data, e.g. monotonicity or convexity by means of the functional data derivatives. The aforementioned problem is addressed with tools of Global Optimization in continuous variables: the time instants are selected to maximize the correlation between the class label and the Support Vector Machine score used for classification. The effectiveness of the proposal is shown in univariate and multivariate datasets.

## 1. Introduction

Real-time information monitoring has become extremely popular thanks to the technological advances witnessed in recent years. Consequently Functional Data Analysis (FDA), [18,38,39], has become an outstanding field in many real-world areas. See [7,10,32,34,40] for applications in physical and chemical processes, spectrometry, meteorology, or speech recognition, among others. FDA can be seen as a generalization of the standard multivariate analysis, where the data are (multivariate) functions of a continuous parameter instead of vectors in a finite dimensional space. A multivariate functional datum is defined as a finite dimensional vector where each component is a univariate function. A simple approach to handle functional data consists of the discretization of the multivariate observed function to then apply multivariate techniques to the resulting vector. Nevertheless, functional data are intrinsically infinite dimensional and have properties that differentiate them from multivariate data. In fact, the direct use of the standard multivariate methodologies shows severe limitations. For instance, multivariate analysis is not able to deal with situations where the curse of dimensionality appears, i.e. when the

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

https://doi.org/10.1016/j.ins.2018.12.060
0020-0255/© 2018 Elsevier Inc. All rights reserved.

number of covariates is larger than the number of individuals. On the other hand, although theoretically functional data are infinite dimensional, in practice they are discretized in a grid of points, yielding high dimensional vectors with larger cardinality than the number of records. Therefore, some issues may appear when solving such type of instances. Furthermore, the strong correlation between two consecutive points of the functions, which is an intrinsic characteristic of functional data, may not be adequately considered in traditional multivariate analysis procedures.

Many efforts have been done in the literature to extend the multivariate techniques to FDA, mostly for univariate functional data, and specific problems such as clustering [24] and PCA [3] Hence, it is desirable to promote the development of new tools which exploit the functional structure of the multivariate functional data.

In this paper, we address the problem of classifying multivariate functional data into two prefixed classes by using the information provided by a training sample [18]. Classifiers will be based on the benchmark supervised classification tool Support Vector Machines (SVM), [32,34,40]. For other classification techniques used with functional data, the reader is referred to [18,37] and [1] for a survey.

Functional data classification entails some difficulties associated with the high computational costs, and the introduction of redundancy and noise from measurement errors, which may deteriorate the correct classification rate. Since functional data are intrinsically infinite dimensional data, it is thus useful to select the time instants providing the most relevant information of the data, i.e., to perform variable selection.

Advantages of the variable selection are highlighted in what follows. First, interpretability may be enhanced and monitoring costs may be reduced if just a few time instants capable of discriminating the behavior of functional data are considered instead of the full time interval. Second, variable selection may lead to a better performance of the classifier. The reduction of dimensionality of the (discretized) functional data may result in a better classification, since the correlation between features (time instants) may negatively affect classification rates.

The feature selection problem has been widely studied in the multivariate data literature. See for instance the surveys [27], the papers [2] addressing classification problems, [33] for regression, and [28] for clustering. Particularly, some references in very high-dimensional problems such as cancer detection via gene expression data, must be mentioned. The work of [36] presents a theoretical and practical framework for feature selection based on a conditional mutual information criterion. [35,50] focus on the chemotherapy effectiveness problems solved by means of ranking (SVM-RFE) and fuzzy if-then rules, respectively. Moreover, the paper of [31] combines a Difference of Convex functions Algorithm (DCA) with a double-regularized SVM formulation to select the most important features.

Dimensionality reduction techniques, based on the projection of the functional data on lower-dimensional spaces have also been considered. These include, among others, Functional Principal Component Analysis (FPCA) [30], Partial Least Squares (PLS) [37], and B-splines functions [47]. For other dimensionality reduction techniques in functional data, see [48]. In this paper, however, we focus on a different approach of dimensionality reduction based on variable selection. Our goal is, as mentioned before, to select the most informative time instants in order to obtain good classification rates. In such a context of variable selection, some regression papers should be mentioned. In [49] for instance, the standard LASSO and Dantzig selector procedures are proposed. Moreover, [25] focuses on the interpretability of the coefficient function, whereas [17] work with nonlinear models.

On univariate functional classification, some multivariate techniques have been directly applied to select the relevant features. This is the case, for example, of [22] where new covariates, e.g. mean, maximum and minimum value, are extracted from the functional data in order to select the most important variables based on a mutual information criterion. In the work in [29], one single time instant is sought. As admitted in the paper, it is not possible to generalize their methodology to search a set of more time instants. With respect to the selection of time instants, we should emphasize the recent works of [4,6,43,44], where greedy approaches, yielding local optima, are used. These papers follow a combinatorial approach: such time instants are assumed to belong to the finite set of instants at which actual measurements exist.

Variable selection for multivariate functional data has not been analyzed in the literature, to the best of our knowledge. Therefore, the main contribution of this paper is to provide a new strategy able to seek the most informative time instants to achieve good classification rates in multivariate functional data. Contrary to what is usually done in the literature, [4,6,43,44], we consider the time as a continuous variable, and we search for the global solution using a surrogate of the number of misclassified data, namely the correlation between the SVM score and the actual class. See [41,44] for a deeper analysis. Finding such optimal time instants amounts to solving a continuous smooth optimization problem. Moreover, our algorithmic strategy is improved thanks to the definition of nested models of increasing complexity, following the idea in [8,11].

Finally, our approach involves a framework which can accommodate from one to several functions, allowing one to address in the very same way univariate and multivariate functional data. In particular, one can easily include in the model high-order information (monotonicity, convexity, ...) by replacing each single-variate functional datum by a multivariate functional datum, corresponding to the functional datum itself and its derivatives. The information provided by the derivatives has been utilized in the clustering context, [23], with outstanding results.

The remainder of this paper is structured as follows. Section 2 explains the details of the SVM model applied to functional data. In Section 3 the optimization method used in our approach, as well as the solving strategy and some improvements of the method, are detailed. Section 4 is focused on the numerical experiments, and finally, Section 5 is devoted to present some conclusions and extensions.

## 2. Variable selection with functional SVM

In this section, the SVM problem for multivariate functional data is introduced. For a deeper analysis of SVM, the reader is referred to [14]. Section 2.1 explains the notation used in this paper and details how the high-order information can be included in the multivariate data structure. In Section 2.2 the basic concepts of the linear and nonlinear SVM are explained. Finally, Section 2.3 is devoted to the details of the kernel function employed along this work.

### *2.1. Some notation and high-order information*

We assume given a sample $s$ of individuals, where each instance $i \in s$ is associated to the pair $(X_i, Y_i)$. The datum $X_i \in \mathcal{X} = \mathcal{F}^p$ is composed by $p$ functional features, i.e. $X_i = (X_{i1}(t), \dots, X_{ip}(t))$, with $X_{iv} : [0, T] \to \mathbb{R}$, $v = 1, \dots, p$ belonging to the class $\mathcal{F}$ of $d-$times continuously differentiable functions on the time interval $[0, T]$. Furthermore, $Y_i \in \{-1, +1\}$ denotes the class label of the observation $i \in s$. Our aim is to find a classification rule which allows us to infer the class $Y$ of a new functional observation $X \in \mathcal{X}$.

It is worthwhile to mention that our methodology is not only restricted to pure multivariate functional data. Indeed, the approach here proposed can be directly applied to univariate functional data, $X(t) \in \mathcal{F}$. More specifically, apart from the straightforward case in which one just considers $p = 1$, we can also make a pre-processing which transforms the univariate data into multivariate by taking advantage of the high-order information throughout the usage of the derivatives of $X$. This process yields data of the form:

$$(X(t), X'(t), \dots, X^{d)}(t)), \tag{1}$$

where $X^{d)}(t)$ denotes the $d$-th derivative of $X(t)$. Moreover, the information provided by the derivatives can also be added to the pure multivariate functional case, yielding

$$(X_1(t), \dots, X_p(t), X_1'(t), \dots, X_p'(t), \dots, X_1^{d)}(t), \dots, X_p^{d)}(t)), \tag{2}$$

The numerical experience in Section 4 shows that the high-order information will be crucial in the classifier performance.

### *2.2. SVM classification for functional data*

Regarding the SVM classification, when the instances in the training sample are linearly separable, SVM provides an optimal hyperplane $\langle \mathbf{w}, X_i \rangle + b$, separating both classes, where $\mathbf{w} \in \mathcal{X}$, $b \in \mathbb{R}$ and $\langle \cdot, \cdot \rangle$ denotes the inner product in the functional space $\mathcal{X}$. Such hyperplane is obtained by maximizing the so called margin, i.e. the distances to the closest positive and negative training data. The maximal margin is provided by the element $\mathbf{w}$ with minimum norm such that $Y_i(\langle \mathbf{w}, X_i \rangle + b) \geq 1$, $\forall i \in s$. Furthermore, since perfect classification of the training sample is quite rare, some classification errors are allowed thanks to the artificial variables $\xi_i$ introduced for all $i \in s$. In that case, the optimal solution of the linear SVM is obtained by solving the following optimization problem:

$$\begin{cases} \min\limits_{\mathbf{w}, b, \xi} & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum\limits_{i \in s} \xi_i \\ \text{s.t.} & Y_i(\langle \mathbf{w}, X_i \rangle + b) \geq 1 - \xi_i, \ i \in s, \\ & \xi_i \geq 0, \quad i \in s \end{cases} \tag{3}$$

The parameter $C$ is a regularization parameter, to be tuned, that penalizes the existence of misclassified observations in the training sample [45]. Larger values of $C$ yield smaller-margin hyperplanes, whilst smaller values of $C$ result in larger-margin hyperplanes, even if they misclassify more data. In order to tune the parameter $C$, $k$-fold crossvalidation with a grid search on a sufficiently large interval is usually applied. See [13] for further information.

The procedure above defines a linear classification rule: given $\mathbf{w}$, optimal solution of (3), a score $\hat{Y}(X) = \langle \mathbf{w}, X \rangle$ is associated to each functional data $X$, and thus $X$ is classified in class $+1$ if and only if $\hat{Y}(X) > \beta$, where $\beta$ is a prefixed threshold value. To gain versatility in the procedure, records are mapped to a higher dimensional space by a nonlinear feature map, and a so-called kernel $K$ is defined in the space of functions $\mathcal{X}$, [45]. Instead of solving (3), the following quadratic concave maximization problem with linear constraints is solved:

$$\begin{cases} \max\limits_{\alpha} & \sum\limits_{i \in s} \alpha_i - \frac{1}{2} \sum\limits_{i,j \in s} \alpha_i \alpha_j Y_i Y_j K(X_i, X_j) \\ \text{s.t.} & \sum\limits_{i \in s} \alpha_i Y_i = 0 \\ & \alpha_i \in [0, C], \ i \in s, \end{cases} \tag{4}$$

This way, a nonlinear classification rule is obtained: given $\alpha$, optimal solution of (4), a score $\hat{Y}(X)$ in (5) is associated with each functional data $X$,

$$\hat{Y}(X) = \sum_{i \in s} \alpha_i Y_i K(X, X_i), \quad X \in \mathcal{X}, \tag{5}$$

and thus $X$ is classified in class $+1$ if and only $\hat{Y}(X) > \beta$.

### 2.3. Kernel function

Several kernel functions have been proposed in the literature for finite dimensional data, e.g., the linear kernel, [14], the polynomial kernel, [34,40], or the Gaussian (RBF) kernel, [11,14].

Let $H$, $p \geq 1$ be integers, and consider a kernel $\widetilde{K}: \mathbb{R}^{Hp} \times \mathbb{R}^{Hp} \longrightarrow \mathbb{R}$. Given a vector $\mathbf{t} = (t_1, \ldots, t_H)$ of $H$ time instants in $[0, T]$, we can define a kernel $K$ for $p-$variate functional data $X_i = (X_{i1}, \ldots, X_{ip}) \in \mathcal{X}$ by taking into account the values of each functional record $X_v$ at time instants $t_1, \ldots, t_H$, namely,

$$K(X_i, X_j, \mathbf{t}) = \widetilde{K}((X_i(t_1), \ldots, X_i(t_H)), (X_j(t_1), \ldots, X_j(t_H))), \quad X_i, X_j \in \mathcal{X}$$

For instance, from the Gaussian kernel $\widetilde{K}$,

$$\begin{aligned}&\widetilde{K}((f_{11}, \ldots, f_{1p}, \ldots, f_{H1}, \ldots, f_{Hp}),\\ &\quad (g_{11}, \ldots, g_{1p}, \ldots, g_{H1}, \ldots, g_{Hp})) = \exp\left(-\sum_{v=1}^{p}\sum_{h=1}^{H}\omega_v (f_{hv} - g_{hv})^2\right)\end{aligned}$$

one obtains the following Gaussian kernel $K$ for $p-$variate functional data

$$\begin{aligned}K(X_i, X_j, \boldsymbol{\omega}, \mathbf{t}) &= \widetilde{K}((X_{i1}(t_1), \ldots, X_{ip}(t_1), \ldots, X_{i1}(t_H), \ldots, X_{ip}(t_H)),\\ &\qquad (X_{j1}(t_1), \ldots, X_{jp}(t_1), \ldots, X_{j1}(t_H), \ldots, X_{jp}(t_H)))\\ &= \exp\left(-\sum_{v=1}^{p}\sum_{h=1}^{H}\omega_v (X_{iv}(t_h) - X_{jv}(t_h))^2\right), \quad X_i, X_j \in \mathcal{X}.\end{aligned} \tag{6}$$

In this paper, for simplicity, we only focus on the Gaussian kernel, since it is one of the most used and effective kernels, but the methodology proposed is easily extended to other classes.

Hence, our objective is to select the time instants that provide the most relevant information for discriminating between the two groups. A global optimization approach for this selection will be proposed in the next sections. Two types of parameters are to be tuned: the vector, $\mathbf{t} = (t_1, \ldots, t_H)$, such that

$$0 \leq t_1 \leq \ldots \leq t_H \leq T \tag{7}$$

and the parameters associated to the SVM problem (4), i.e. the regularization parameter $C$ and the bandwidth $\boldsymbol{\omega}$. Extra constraints over the parameters can be easily imposed in the optimization problem. For instance, if we want the $H$ time instants to be far apart from each other, we may add the constraints

$$t_{h+1} \geq t_h + \delta, \quad h = 1, \ldots, H-1$$

for some fixed $\delta > 0$.

Following the methodology of [8], we propose in this paper to combine a grid search to tune the parameter $C$ and an alternating procedure to seek the bandwidth $\boldsymbol{\omega}$, and the time instants $\mathbf{t}$.

Details about the resulting optimization problem and the solving strategy are given in Section 3.

## 3. A global optimization approach

In this section, the mathematical formulation of the variable selection problem in SVM classification with functional data is presented Section 3.1 is devoted to the problem formulation and how to solve such a problem, whereas a nested heuristic is proposed in Section 3.2, in which we take advantage of the fact that the different time instants $\mathbf{t} = (t_1, \ldots, t_H)$ can be easily embedded in a nested structure of models. Section 3.3 addresses the problem of determining the number $H$ of time instants.

### 3.1. The bilevel optimization problem

As previously mentioned, two different types of decision variables are involved in the variable selection problem for classification of functional data with SVM. First, the $H$ time instants $\mathbf{t} = (t_1, \ldots, t_H)$ satisfying (7), and second, the parameters $C$ and $\boldsymbol{\omega}$ involved in the SVM problem (4), and in the Gaussian kernel (6), respectively.

A strategy analogous to the one used in [8] is proposed to find the optimal values of $C$, $\boldsymbol{\omega}$ and $\mathbf{t}$. $C$ is obtained by using a standard grid search, while a bilevel optimization problem is defined to tune the parameters $\boldsymbol{\omega}$ and $\mathbf{t}$. In such bilevel problem, we propose to maximize the Pearson correlation between the class label $Y_i$ of the observation $i \in s$, and the score $\hat{Y}(X_i(\mathbf{t}), \boldsymbol{\omega}, \alpha)$ due to two main reasons. Firstly, this surrogate has been recently proposed in [8,41,44] with outstanding results, and secondly, such dependency measure yields a smooth optimization problem, in which gradient information can be used to speed up convergence. This last issue means a significant advantage over the use of other performance measures, such as those based on the confusion matrix, which usually leads to mixed-integer optimization problems hard to solve for realistic data sizes.

It is known that variable selection and parameter tuning may lead to overfitting when the optimization is directly performed in the whole dataset, Chapter 7 of [21]. To avoid overfitting and obtain more stable solutions, frequently the data are randomly divided into training, validation and testing samples. This process is repeated $k$ times by performing $k-$fold cross-validation.

In this paper, the parameters and time instants sought, as well as the performance estimates of the classifier, are obtained as follows: the database is split into $k$ folds. Then, $k-1$ folds are chosen to be again divided into three parts, yielding the samples $s_1$, $s_2$ and $s_3$. Finally, the remaining fold constitutes the fourth independent sample $s_4$. Samples $s_1$ and $s_2$ act training samples, while $s_3$ and $s_4$ are the validation and testing samples, respectively. This division process is repeated one time per fold.

Regarding the role of each sample in the optimization strategy, sample $s_1$ is used to obtain the SVM dual variables, $\alpha$, solving Problem (4) for fixed $\boldsymbol{\omega}$**,** $\mathbf{t}$ and $C$. Sample $s_2$ is employed to compute $R((Y_i, \hat{Y}(X_i(\mathbf{t}, \boldsymbol{\omega}, \alpha))_{i\in s_2})$, i.e. the correlation between the class labels and the scores. Sample $s_3$ is used to tune the regularization parameter $C$, by evaluating the accuracy for all the values of $C$ in a grid, and keeping the one with the largest value. Finally, the accuracy obtained with the optimal parameters is estimated on the independent sample $s_4$.

To sum up, for a fixed $C$, the resulting bilevel optimization problem is given in (8)

$$\begin{cases} \max\limits_{\boldsymbol{\omega},\mathbf{t}} & R((Y_i, \hat{Y}(X_i(\mathbf{t}), \boldsymbol{\omega}, \alpha))_{i\in s_2}) \\ \text{s.t.} & \alpha \text{ solves (4) in } s_1, \\ & \omega_v \geq 0, v = 1, \ldots, p, \\ & 0 \leq t_1 \leq \cdots \leq t_H \leq T \end{cases} \tag{8}$$

Note also that we have emphasized the dependence of the score $\hat{Y}$ on the time instants in $\mathbf{t}$, on the bandwidth $\boldsymbol{\omega}$, and on the classification coefficients $\alpha$ in the notation. When such values are clear, they will be omitted in the notation for the sake of simplicity.

Problem (8) is a nonlinear problem which can be solved with the techniques described in e.g. [12]. For instance, we may mention branch-and-bound schemes in which the problem is reformulated under some convexity assumptions using the Karush–Kuhn–Tucker (KKT) conditions. Even with these reductions, the so-obtained problem is difficult to solve due to the nonconvexities in the complementary and Lagrangian constraints. Penalty function methods can also be used to solve bilevel problems. Convergence, is however, to stationary points.

Instead, we propose to address the bilevel problem (8) for each $C$ by a procedure consisting in two alternating steps: the SVM step, in which for $\boldsymbol{\omega}$ and $\mathbf{t}$ fixed, we solve Problem (4) to obtain the optimal SVM variables $\alpha$; and the max-corr step, where for $\alpha$ fixed, one maximizes the Pearson correlation $R$ in (9) to obtain the optimal bandwidth $\boldsymbol{\omega}$ and the time instants $\mathbf{t}$. This correlation maximization problem can be expressed as:

$$\begin{cases} \max\limits_{\boldsymbol{\omega},\mathbf{t}} & R((Y_i, \hat{Y}(X_i(\mathbf{t}), \boldsymbol{\omega}))_{i\in s_2}) \\ \text{s.t.} & \omega_v \geq 0,\ v = 1, \ldots, p, \\ & 0 \leq t_1 \leq \cdots \leq t_H \leq T \end{cases} \tag{9}$$

Different strategies are used to solve Problems (4) and (9). The SVM problem (4) is a quadratic concave maximization problem with linear constraints. Therefore, standard local search routines or specific tools, as in [19], can be applied. On the other hand, Problem (9) is a continuous optimization problem, where classic local searches are combined with a multistart approach to avoid getting stuck at local optima.

The initial values of $\boldsymbol{\omega}$ and $\mathbf{t}$ in the first iteration of the alternating approach are randomly selected in their corresponding domains of definition.

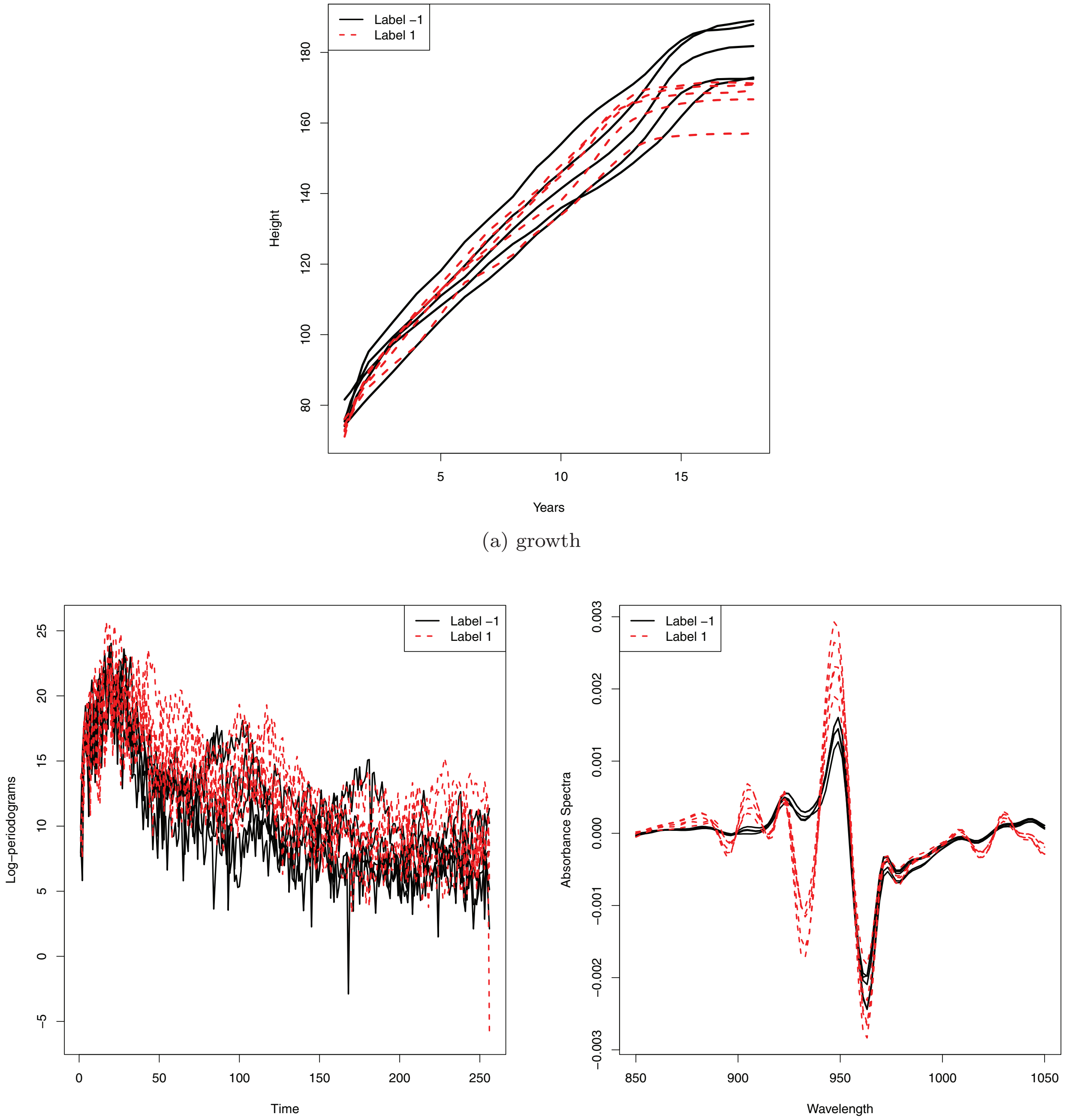
The alternating procedure is running until some stopping criteria, such as the number of evaluations or the maximum time allowed is reached, yielding certain values of $\boldsymbol{\omega}$**,** $\mathbf{t}$ and $\alpha$, for a fixed $C$. The value of $C$ is chosen by applying a grid search, i.e. for each value of $C$ in a grid, the accuracy obtained with the classification rule that the parameters gives after solving Problem (8), is measured in sample $s_3$. The parameter $C$ with the largest value in terms of accuracy will be kept.

Finally, we test our approach by measuring the accuracy in a forth sample, $s_4$.

It is worth-mentioning that calculating the gradient of the objective function in (9) will reduce the computational effort, since numerical differentiation is avoided. Just applying the chain rule and taking into account (10) we can easily obtain an explicit expression for the gradient of the objective function in (9):

$$\begin{aligned} \frac{\partial K\big(X_i, X_j, \omega, \mathbf{t}\big)}{\partial \omega_v} &= K\big(X_i, X_j, \omega, \mathbf{t}\big)\left(-\sum_{h=1}^{H}\big(X_{iv}(t_h) - X_{jv}(t_h)\big)^2\right),\ v = 1, \ldots, p \\ \frac{\partial K\big(X_i, X_j, \omega, \mathbf{t}\big)}{\partial t_h} &= -2\,K\big(X_i, X_j, \omega, \mathbf{t}\big)\sum_{v=1}^{p}\big(\omega_v\big(X_{iv}(t_h) - X_{jv}(t_h)\big)\big) \times \\ &\quad \times\left(\frac{\partial X_{iv}(t)}{\partial t}|_{t=t_h} - \frac{\partial X_{jv}(t)}{\partial t}|_{t=t_h}\right),\ h = 1, \ldots, H \end{aligned} \tag{10}$$

We recall that, in practice, the original functional data $X_i$ may be only available throughout a grid of time instants. Therefore, interpolation techniques, such as cubic splines, [15,21], should be used as a preprocessing step so that the functional



(a) growth

(b) phoneme

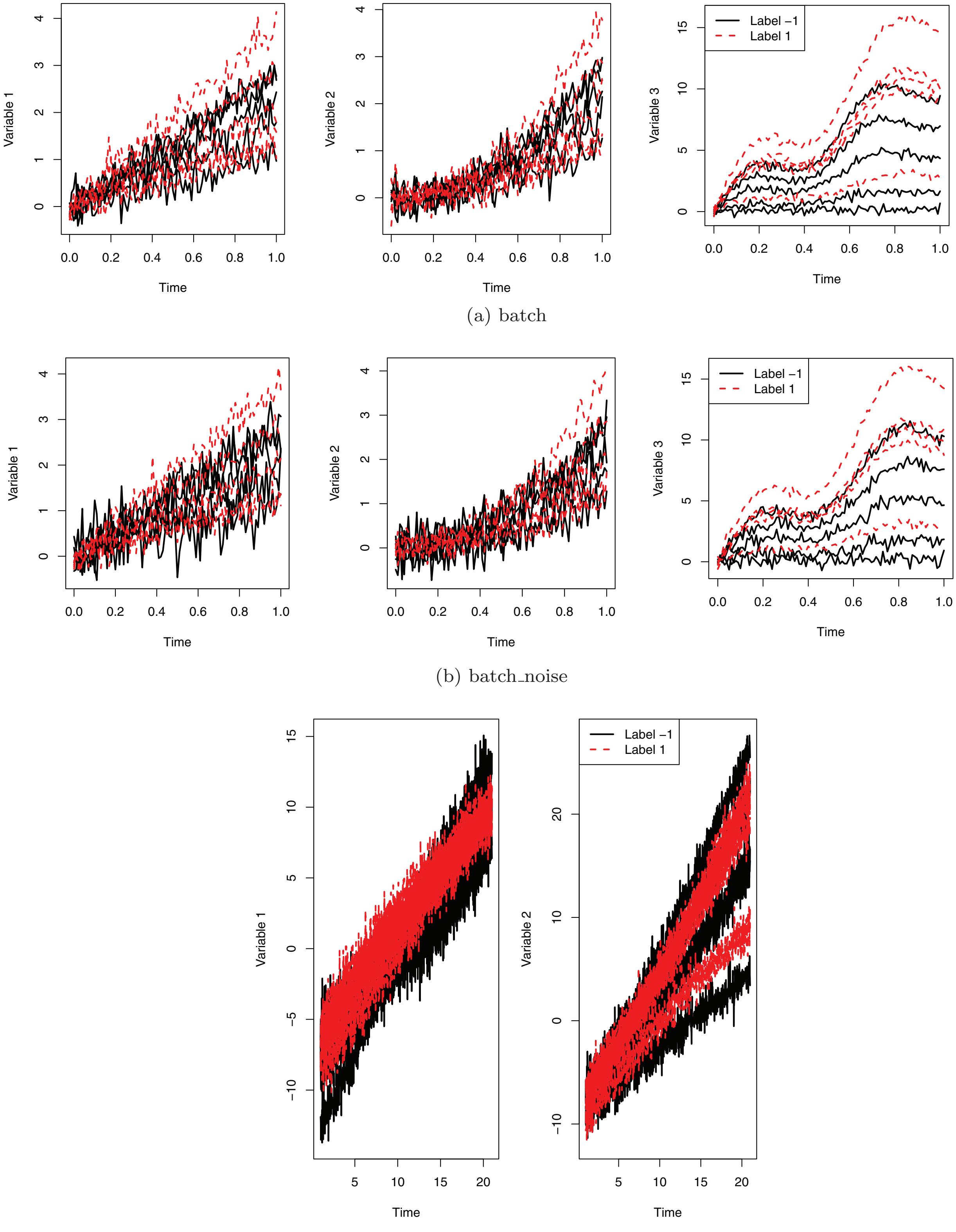
(c) tecator

**Fig. 1.** Sample of functional data in the univariate datasets analyzed.

data can be properly rebuilt. It is important to remark that the interpolation step recovers the smoothness of the data with respect to $\boldsymbol{\omega}$, $\mathbf{t}$.

Furthermore, if we want to take advantage of the high-order information of the data, it is necessary to get, as pre-processing, the derivatives from the data $X(t)$. One possible choice would be to compute the derivative of the smoothed data. Nevertheless, in order to avoid numerical errors from the interpolation, we suggest using the finite-increments as an approximation of the derivatives. For instance, if the first derivative of $X(t)$ in the point $t_h$ should be computed, one has:

$$X'(t_h) = \frac{X(t_h) - X(t_{h-1})}{t_h - t_{h-1}} \tag{11}$$

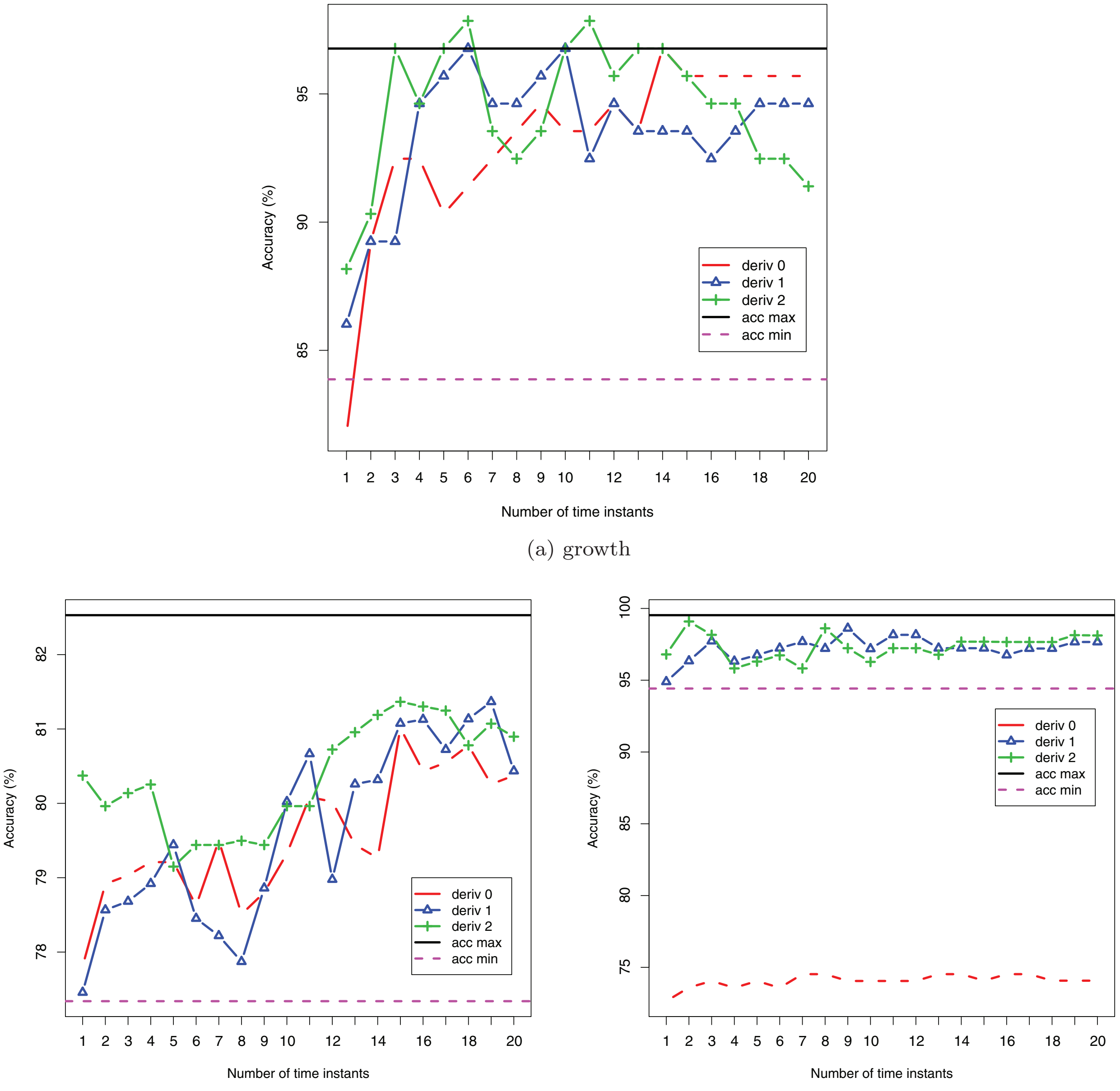


(a) batch

(b) batch_noise

(c) trigonometric

**Fig. 2.** Sample of functional data in the multivariate datasets analyzed.



(a) growth

(b) phoneme

(c) tecator

**Fig. 3.** Average accuracy in the univariate datasets analyzed.

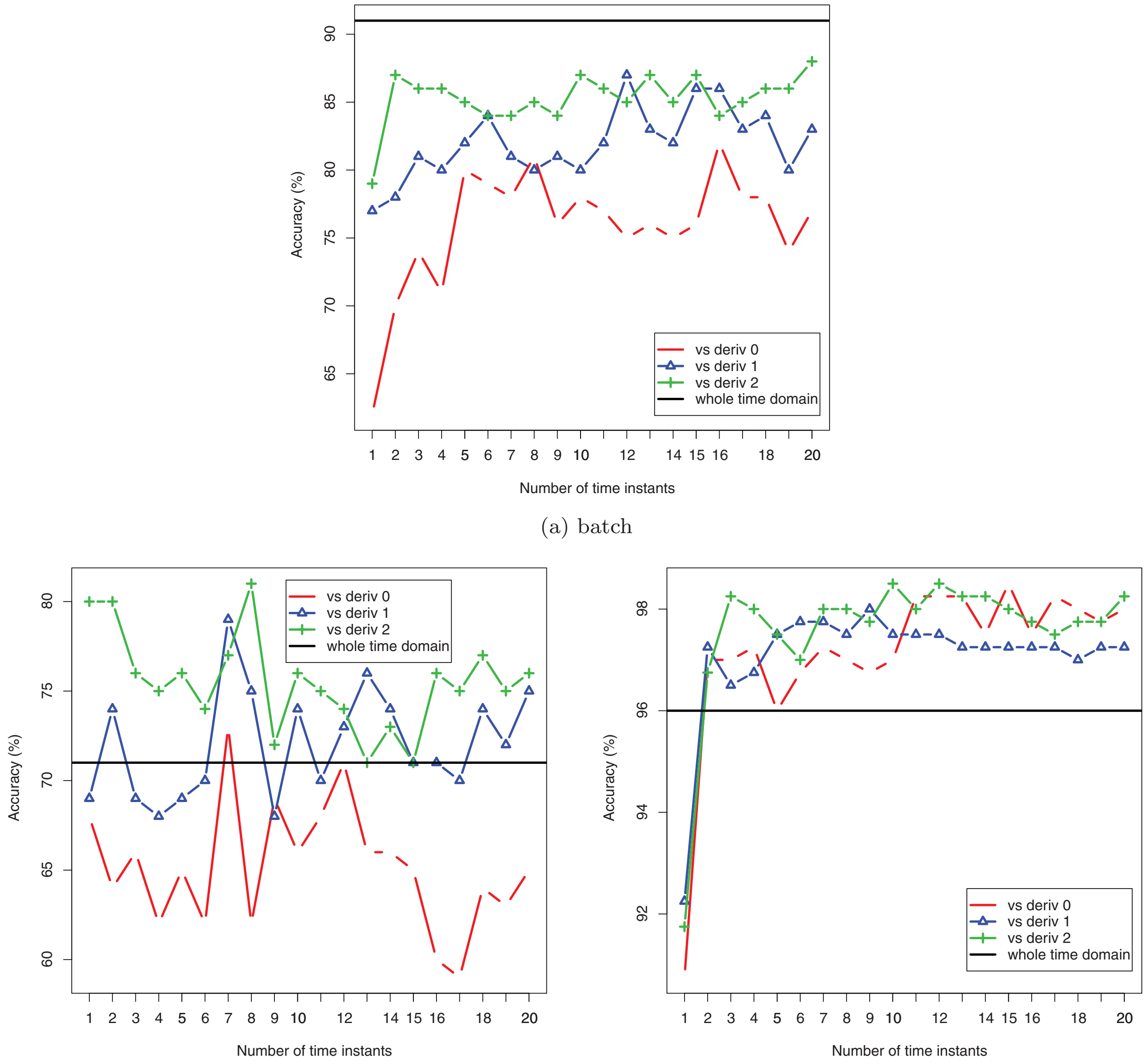
Note that in (11), $t_h, \forall h$, indicates the time instants where the functional data are discretized. The formula in (11) should be reproduced for all the time points of the discretization, and it is easy to see that it can be extended to any derivative's order. After obtaining the discretized derivatives, they should be smoothed with an interpolation technique, as explained before.

A pseudocode of our approach is outlined in Algorithm 1, and an extension of it based on a nested heuristic is detailed in Section 3.2.

### 3.2. A Nested Heuristic

In this section we enhance the basic heuristic detailed in Algorithm 1. Adopting the same idea of [8,11], we propose to define a series of nested models of increasing complexity, where the optimal solution of the elementary case is used as a starting solution in the following more complex model.

The idea is that, in order to find the vector $\mathbf{t}^{h+1}$ of $h+1$ time instants, one can use as starting solution a perturbation of $\mathbf{t}^h$, the solution obtained when only $h$ time instants are sought. Therefore, if we want to find the $H$ time instants which



(a) batch

(b) batch_noise

(c) trigonometric

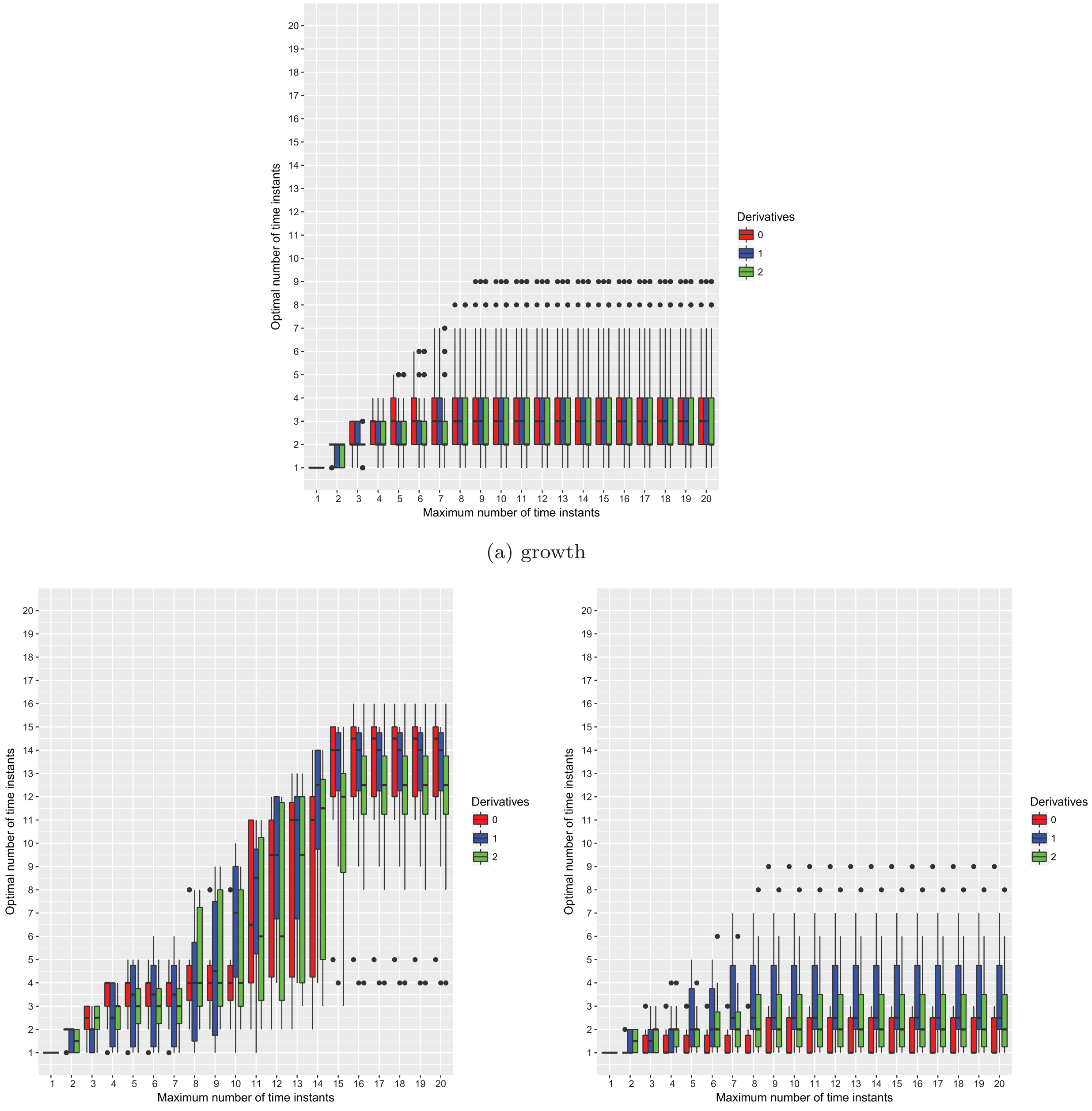
**Fig. 4.** Average accuracy in the multivariate datasets analyzed.

best discriminate between two groups, we solve successively the Alternating Procedure of Algorithm 1 for $h=1$ to $H$, but considering the easy-to-tune structure of the simple models as a simplification of the complex cases, in such a way that the (suboptimal) solution $K(X_i, X_j, \boldsymbol{\omega}^h, \mathbf{t}^h)$ is used as initial solution for kernel $K(X_i, X_j, \boldsymbol{\omega}^{h+1}, \mathbf{t}^{h+1})$. More precisely, in order to build the initial solution for the $h+1$ time instants in $\mathbf{t}^{h+1}$, we first select a random value $\tau \in [0, T]$, and then we include it in the appropriate position of the optimal solution of the level $h$, $\mathbf{t}^h_{opt}$, in such a way that $\mathbf{t}^{h+1}$ satisfies the conditions in (7), i.e. $\mathbf{t}^{h+1} := \sigma(\tau, \mathbf{t}^h_{opt})$, where $\sigma$ is the function that sorts in increasing order the time instants $\mathbf{t}^h_{opt}$ and $\tau$.

One of the advantages of our nested heuristic is that it allows us to obtain a trajectory of the accuracy in terms of the number of time instants chosen. This is a crucial issue, since, in practice, the number $H$ of time instants to consider may not be fixed, and thus a list of classifiers, with different complexity ($H$) and accuracy, can be provided.

Note that the solution of the level $h$ will be used just as starting point of level $h+1$, in order to speed the algorithm, but still allows the algorithm to yield a solution that is very different from the level $h$ solution. In this way, our proposal clearly differs from [44], where greedy schemes are proposed.

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



(a) growth

(b) phoneme

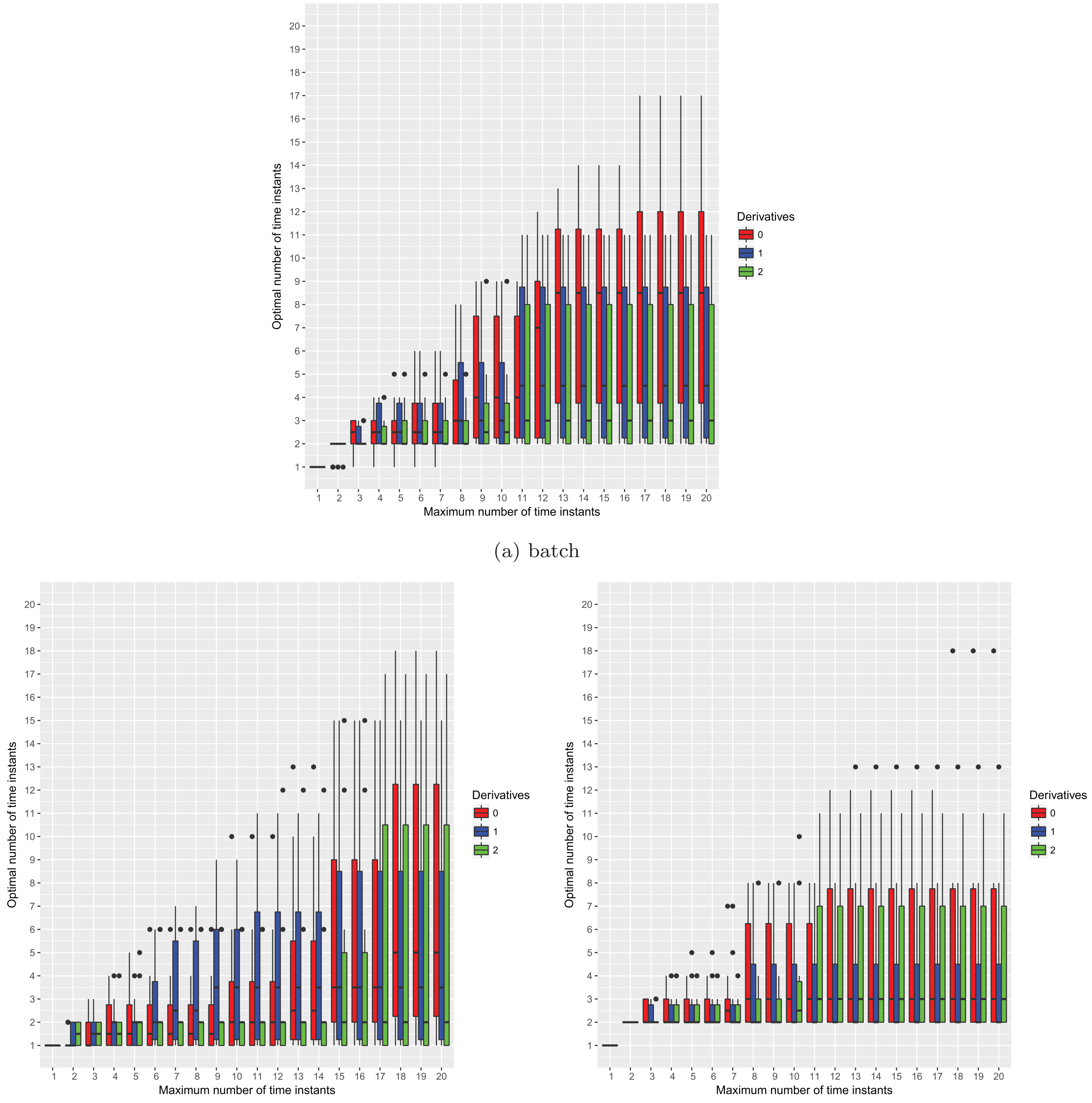
(c) tecator

**Fig. 5.** Boxplots of the optimal number of time instants in the univariate datasets.

### 3.3. *Choice of the number of variables,* H

The choice of the optimal number of time instants, $H$, is a critical issue. The larger is $H$, the better is the classification accuracy expected to be obtained, although the risk of overfit increases. However, the smaller the value of $H$, the easier the interpretation of the results obtained.

In this paper we propose to follow the common strategy carried out in the literature, [4,6,43,44], and choose the value of $H$ by estimating the accuracy on the validation sample $s_3$ with $k-$fold crossvalidation. The value of $H$ with the largest accuracy will be kept.



(a) batch

(b) batch_noise

(c) trigonometric

**Fig. 6.** Boxplots of the optimal number of time instants in the multivariate datasets.

## 4. Numerical experiments

This section details the computational results of our approach, in which we provide the accuracy obtained when only some selected time instants and not the whole functional interval $[0, T]$ is considered. Section 4.1 describes the settings of the computational experience. The results obtained on the different databases are presented in Section 4.2.

### *4.1. Description of the experiments*

Our proposal has been applied to both univariate and multivariate functional data. On top of comparing the performance of the SVM based on the full time interval against the SVM classifier for data measured at just $H$ time instants, we have also analyzed the improvements in performance obtained when instead of the functional data alone, up to $d$ derivatives of

**Table 1**
Data description summary.

| | #records | #time instants | #records label -1 | #records label +1 | #components |
|---|---|---|---|---|---|
| growth | 93 | 31 | 54 | 39 | 1 |
| phoneme | 1717 | 256 | 1022 | 695 | 1 |
| tecator | 215 | 100 | 77 | 138 | 1 |
| batch | 100 | 101 | 50 | 50 | 3 |
| batch_noise | 100 | 101 | 50 | 50 | 3 |
| trigonometric | 400 | 1001 | 200 | 200 | 2 |

the functional data are also included in the input. For this reason, we have also run Algorithm 2 for three different values of $d$, namely $d = 0, 1, 2$, which correspond respectively to the cases in which just the information of the functional data, or also its monotonicity, or both monotonicity and convexity, are considered.

In order to obtain stable results, $k-$fold cross-validation is performed. The number of folds, $k$, depends on the dataset considered. Particularly, if a database is small, $k$ coincides with the number of individuals, that is to say, leave-one-out is applied. On the other hand, for big databases $k = 10$ has been chosen. In this paper, we consider that a dataset is small if its cardinal is smaller than 100 individuals. See Table 1. Algorithm 2 is run $k$ times, one per fold. Each time, the dataset is

**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 $C$ in the grid do
    Alternating Procedure
    repeat
        1. Fixed $\boldsymbol{\omega}, \mathbf{t}$, calculate the parameters $\alpha$ of the SVM clasiffier by solving Problem (4) using $s_1$.
        2. Fixed $\alpha$, calculate $\boldsymbol{\omega}, \mathbf{t}$ by solving Problem (9) over $s_2$.
    until stopping criteria
    • Evaluate the accuracy using the sample $s_3$ for the $C$ fixed in the grid.
end for
• The optimal value of $C$ is the one with best accuracy in $s_3$, and the optimal values of $\alpha$, $\boldsymbol{\omega}$ and $\mathbf{t}$ are the parameters
associated to the optimal $C$.
Output: Optimal parameters $\boldsymbol{\omega}, \mathbf{t}, C, \alpha$, and the accuracy estimated from $s_4$.
```

**Algorithm 2** Nested heuristic for variable selection.

```
Input: $H$, nested kernels $K(X_i, X_j, \boldsymbol{\omega}^1, \mathbf{t}^1) \prec \ldots \prec K^H(X_i, X_j, \boldsymbol{\omega}^H, \mathbf{t}^H)$.
• 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 $C$ in the grid do
    Initialization:
    • $h := 1$.
    • Randomly select an initial solution $\tilde{\boldsymbol{\omega}}^1 \in [0, +\infty)^p$ and $\tilde{\mathbf{t}}^1 := t_1 \in [0, T]$.
    • Set $(\boldsymbol{\omega}, \mathbf{t}) := (\tilde{\boldsymbol{\omega}}^1, \tilde{\mathbf{t}}^1)$.
    while $h \le H$ do
        1. Run the Alternating Procedure of Algorithm 1 for $K(X_i, X_j, \boldsymbol{\omega}^h, \mathbf{t}^h)$, starting from $(\boldsymbol{\omega}, \mathbf{t})$ and yielding$(\boldsymbol{\omega}^h_{opt}, \mathbf{t}^h_{opt})$ as
           solution, using samples $s_1$ and $s_2$.
        2. Randomly generate $\tau \in [0, T]$.
        3. Set $\boldsymbol{\omega}^{h+1} := \boldsymbol{\omega}^h_{opt}$, $\mathbf{t}^{h+1} := \sigma(\tau, \mathbf{t}^h_{opt})$, $(\boldsymbol{\omega}, \mathbf{t}) := (\boldsymbol{\omega}^{h+1}, \mathbf{t}^{h+1})$ and $h := h + 1$.
        4. Evaluate the accuracy over the sample $s_3$with $C$ fixed.
    end while
end for
• For $h$ fixed, the optimal value of $C$ is the one with the best accuracy in $s_3$. The optimal values of $\alpha$, $\boldsymbol{\omega}$ and $\mathbf{t}$ are the
parameters associated to the optimal $C$.
Output: Optimal parameters $\boldsymbol{\omega}^h_{opt}, \mathbf{t}^h_{opt}$, $\forall h$, the associated coefficients $C, \alpha$,
and the accuracy estimated from $s_4$.
```

divided into four samples $s_1 - s_4$ as explained in Section 3.1. To test our results we provide the average of the accuracy across the folds, measured on $s_4$. The number of iterations of the multistart is five, the number of iterations of the Alternating Procedure is eight, and the (maximum) number of time instants to be selected, i.e., the number of nested kernels is $H = 20$. Finally, the parameter $C$ takes values in the set $\{2^{-10}, \dots, 2^{10}\}$ in logarithmic scale.

Apart from the experiments explained above, we have also tuned the optimal number of time instants, $H$ by performing cross-validation on sample $s_3$, as explained in Section 3.3.

The whole computational experience is executed on a cluster with 2 terabytes of RAM memory at 6.2 TFlops, running CentOS Linux 7.3, and it is coded in $\mathbb{R}$, [42].

### 4.2. Numerical results

Three univariate (Section 4.2.1) and three multivariate (Section 4.2.2) functional databases have been considered to check the performance of our approach. Table 1 shows the number of records of each database, the number of time instants in which the records are measured, the number of records of each class and the number of components of the functional data vector.

Samples of ten individuals of each dataset are plotted in Fig. 1 (univariate data) and 2 (multivariate data). The records in class $-1$ are depicted with a solid black line, whereas the records in class $+1$ are plotted in red dashed line. Section 4.2.3 is devoted to the computational experience for the optimal choice of the number of time instants to be considered, $H$.

#### 4.2.1. Results on univariate functional data

First, our methodology is tested in three databases widely used in the literature, namely *growth*, [34,44], *phoneme*, [5,20,21], and *tecator*, [18,32,40,44]. Table 2 reports the averaged accuracy on the testing sample provided by Algorithm 2 with the information given by the data ($d = 0$), the first derivative ($d = 1$), and the first two derivatives ($d = 2$). Leave-one-out is performed on the *growth* dataset, whereas 10−fold cross-validation is done in *phoneme* and *tecator*. Our results are compared with *acc max* and *acc min*, respectively the best and worst accuracy results obtained with the state-of-the-art methods, as reported in Tables 2 and 3 of [5].

The same information shown in Table 2 is depicted in Fig. 3. Particularly, the solid red-circled, blue-triangled and green-crossed lines indicates the averaged accuracy obtained with $d = 0, 1, 2,$ respectively. The horizontal black solid line marks the value *acc max*, whereas the horizontal pink dashed line illustrates the value *acc min*.

Two main conclusions are obtained from our analysis. First, our results are competitive against the state-of-the-art. Moreover, the use of high-order information deeply affects the classification performance. This fact is extremely noticeable in the *tecator* dataset. Furthermore, in such database we are very close to the value *acc max* with just $H = 2$ time instants and $d = 2$. If we focus on the *growth* dataset, we realize that with $H = 3, 5, 10, 13, 14$ and $d = 2$ we achieve the same accuracy as the value *acc max*. This fact also happens with $H = 6$ or $H = 10$ and $d = 1$. Furthermore, our methodology is capable of improving the value *acc max* if $H = 6$ or $H = 11$ time instants and $d = 2$ derivatives are considered.

#### 4.2.2. Results on multivariate functional data

Three databases have been analyzed in this section, denoted by *batch*, Section 4.1 of [46], *batch_noise*, Section 4.2 of [46], and *trigonometric*, Section 5.2.2 of [24]. Note that the *trigonometric* dataset is used in [24] for clustering purposes with three and five groups. Nevertheless, in our paper, since binary classification is studied, we only consider two groups. Furthermore, the authors in [46] take the lower bound of the time domain as zero and the upper bound is sampled from a uniform distribution on [0.9, 1.1]. For the sake of simplicity, we assume that the time interval considered in the datasets *batch* and *batch_noise* is [0, 1].

Since, to the best of our knowledge, there is no methodology in the literature which handles the variable selection problem in classification with multivariate functional data, in this section, we compare our results with the standard SVM-classification in which the whole time domain and just the information of the functional data are considered, i.e. $d = 0$. More specifically, we run the SVM problem (4) for the $C$ values in $\{2^{-10}, \dots, 2^{10}\}$, and $\omega_\nu \in \{2^{-5}, \dots, 2^5\}$, for $\nu = 1, \dots, p,$ to then keep the best accuracy as reference value. Both standard SVM and Algorithm 2 have been run using 10−fold cross-validation in all the datasets.

Table 3 and Fig. 4 give the accuracy values of our method for $d = 0, 1, 2,$ plotted in solid red-circled, blue-triangled and green-crossed lines, respectively. Furthermore, the classification accuracy with all the time instants is depicted using a horizontal solid black line.

As in the analysis of univariate functional data, using derivatives turns out to be crucial to enhance classification rates. Moreover, classifying using the information of the whole time interval yields worse accuracy than using only carefully selected time instants. This can be seen, for instance, in the *batch_noise* dataset, where for $H = 7$ and $d = 0,$ accuracy is improved in around two points, or even better with $H = 8,$ and $d = 2,$ where the difference is about ten points. When $d = 2$ derivatives are considered, the accuracy values here obtained are always much better by optimally selecting from $H = 1$ to $H = 20$ than when the whole time domain is taken into account. Focusing on the *trigonometric* dataset, the accuracy values are better when more than $H = 2$ time points are chosen than when the whole time interval is considered.

**Table 2**
Accuracy results on univariate datasets.

| *growth* | | | *H* | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| acc min | acc max | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 83.87 | 96.77 | 0 | 81.72 | 89.24 | 92.47 | 92.47 | 90.32 | 91.39 | 96.52 | 93.54 | 96.76 | 93.54 | 93.54 | 94.62 | 93.54 | 96.77 | 95.69 | 95.69 | 95.69 | 95.69 | 95.69 | 95.69 |
| | | 1 | 86.02 | 89.24 | 89.24 | 94.62 | 95.69 | 96.77 | 94.62 | 94.62 | 95.69 | 96.77 | 92.47 | 94.62 | 93.54 | 93.54 | 93.54 | 92.47 | 93.54 | 94.62 | 94.62 | 94.62 |
| | | 2 | 88.17 | 90.32 | 96.77 | 94.62 | 96.77 | 97.84 | 93.54 | 92.47 | 93.54 | 96.77 | 97.84 | 95.69 | 96.77 | 96.77 | 95.69 | 94.62 | 94.62 | 92.47 | 92.47 | 91.39 |
| *phoneme* | | | *H* | | | | | | | | | | | | | | | | | | | |
| acc min | acc max | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 77.34 | 82.53 | 0 | 77.81 | 78.91 | 79.03 | 79.20 | 79.20 | 78.97 | 79.03 | 78.50 | 78.80 | 79.32 | 80.08 | 80.02 | 79.44 | 79.26 | 81.01 | 80.43 | 80.54 | 80.77 | 80.25 | 80.37 |
| | | 1 | 77.45 | 78.56 | 78.68 | 78.92 | 79.44 | 78.45 | 78.21 | 77.87 | 78.86 | 80.02 | 80.66 | 78.97 | 80.25 | 80.31 | 81.07 | 81.12 | 80.72 | 81.13 | 81.36 | 80.43 |
| | | 2 | 80.37 | 79.96 | 80.13 | 80.25 | 79.14 | 79.44 | 79.44 | 79.49 | 79.44 | 79.96 | 79.96 | 80.72 | 80.95 | 81.18 | 81.36 | 81.30 | 81.24 | 80.78 | 81.07 | 80.89 |
| *tecator* | | | *H* | | | | | | | | | | | | | | | | | | | |
| acc min | acc max | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 94.42 | 99.53 | 0 | 72.64 | 73.59 | 74.04 | 73.57 | 74.04 | 73.57 | 73.57 | 74.52 | 74.04 | 74.04 | 74.04 | 74.04 | 74.52 | 74.52 | 74.06 | 74.52 | 74.52 | 74.06 | 74.06 | 74.06 |
| | | 1 | 94.89 | 96.34 | 97.72 | 96.32 | 96.75 | 97.22 | 97.68 | 97.20 | 98.61 | 97.18 | 98.16 | 98.16 | 97.22 | 97.22 | 97.22 | 96.75 | 97.20 | 97.20 | 97.66 | 97.66 |
| | | 2 | 96.79 | 99.09 | 98.16 | 95.82 | 96.29 | 96.73 | 95.82 | 98.61 | 97.22 | 96.27 | 97.22 | 97.22 | 96.77 | 97.68 | 97.68 | 97.66 | 97.66 | 97.66 | 98.13 | 98.11 |

**Table 3**
Accuracy results on multivariate datasets.

| *batch* | | *H* | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| whole time domain | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 83.87 | 0 | 62.00 | 70.00 | 74.00 | 71.00 | 80.00 | 79.00 | 78.00 | 81.00 | 76.00 | 78.00 | 77.00 | 75.00 | 76.00 | 75.00 | 76.00 | 82.00 | 78.00 | 78.00 | 74.00 | 77.00 |
| | 1 | 77.00 | 78.00 | 81.00 | 80.00 | 82.00 | 84.00 | 81.00 | 80.00 | 81.00 | 80.00 | 82.00 | 87.00 | 83.00 | 82.00 | 86.00 | 86.00 | 83.00 | 84.00 | 80.00 | 83.00 |
| | 2 | 79.00 | 87.00 | 86.00 | 86.00 | 85.00 | 84.00 | 84.00 | 85.00 | 84.00 | 87.00 | 86.00 | 85.00 | 87.00 | 85.00 | 87.00 | 84.00 | 85.00 | 86.00 | 86.00 | 88.00 |
| *batch_noise* | | *H* | | | | | | | | | | | | | | | | | | | |
| whole time domain | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 71.00 | 0 | 68.00 | 64.00 | 66.00 | 62.00 | 65.00 | 62.00 | 73.00 | 62.00 | 69.00 | 66.00 | 68.00 | 71.00 | 66.00 | 66.00 | 65.00 | 60.00 | 59.00 | 64.00 | 63.00 | 65.00 |
| | 1 | 69.00 | 74.00 | 69.00 | 68.00 | 69.00 | 70.00 | 79.00 | 75.00 | 68.00 | 74.00 | 70.00 | 73.00 | 76.00 | 74.00 | 71.00 | 71.00 | 70.00 | 74.00 | 72.00 | 75.00 |
| | 2 | 80.00 | 80.00 | 76.00 | 75.00 | 76.00 | 74.00 | 77.00 | 81.00 | 72.00 | 76.00 | 75.00 | 74.00 | 71.00 | 73.00 | 71.00 | 76.00 | 75.00 | 77.00 | 75.00 | 76.00 |
| *trigonometric* | | *H* | | | | | | | | | | | | | | | | | | | |
| whole time domain | *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 96.00 | 0 | 90.75 | 97.00 | 97.00 | 97.25 | 96.00 | 96.75 | 97.25 | 97.00 | 96.75 | 97.00 | 98.25 | 98.25 | 98.25 | 97.50 | 98.50 | 97.50 | 98.25 | 98.00 | 97.75 | 98.00 |
| | 1 | 91.25 | 97.25 | 96.5 | 96.75 | 97.50 | 97.75 | 97.75 | 97.50 | 98.00 | 97.50 | 97.50 | 97.50 | 97.25 | 97.25 | 97.25 | 97.25 | 97.25 | 97.00 | 97.25 | 97.25 |
| | 2 | 91.75 | 96.75 | 98.25 | 98.00 | 97.50 | 97.00 | 98.00 | 98.00 | 97.75 | 98.50 | 98.00 | 98.50 | 98.25 | 98.25 | 98.00 | 97.75 | 97.50 | 97.75 | 97.75 | 98.25 |

**Table 4**
Average results of the optimal number of time instants on univariate and multivariate databases.

| *batch* | *H* | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 1.80 | 2.40 | 2.60 | 2.70 | 3.00 | 3.00 | 3.70 | 4.90 | 4.90 | 4.90 | 6.50 | 7.60 | 7.70 | 7.70 | 7.70 | 8.50 | 8.50 | 8.50 | 8.50 |
| 1 | 1.00 | 1.90 | 2.30 | 2.80 | 2.80 | 3.00 | 3.00 | 4.00 | 4.10 | 4.10 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 | 5.70 |
| 2 | 1.00 | 1.90 | 2.20 | 2.40 | 2.70 | 2.70 | 2.70 | 2.70 | 3.40 | 3.40 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 |
| *batch_noise* | *H* | | | | | | | | | | | | | | | | | | | |
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 1.20 | 1.60 | 2.00 | 2.10 | 2.20 | 2.20 | 2.20 | 2.20 | 3.10 | 3.10 | 3.10 | 4.30 | 4.30 | 5.70 | 5.70 | 5.70 | 7.40 | 7.40 | 7.40 |
| 1 | 1.00 | 1.40 | 1.60 | 1.90 | 1.90 | 2.80 | 3.30 | 3.30 | 4.00 | 4.00 | 4.50 | 4.50 | 4.50 | 4.50 | 5.40 | 5.40 | 5.40 | 5.40 | 5.40 | 5.40 |
| 2 | 1.00 | 1.50 | 1.50 | 1.70 | 2.10 | 2.40 | 2.40 | 2.40 | 2.40 | 2.40 | 2.40 | 2.40 | 2.40 | 2.40 | 4.30 | 4.30 | 5.80 | 5.80 | 5.80 | 5.80 |
| *growth* | *H* | | | | | | | | | | | | | | | | | | | |
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 1.91 | 2.43 | 2.70 | 2.97 | 3.27 | 3.44 | 3.48 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 |
| 1 | 1.00 | 1.64 | 2.17 | 2.40 | 2.53 | 2.74 | 2.89 | 2.89 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 | 2.94 |
| 2 | 1.00 | 1.70 | 2.06 | 2.23 | 2.43 | 2.56 | 2.68 | 2.89 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 | 3.12 |
| *phoneme* | *H* | | | | | | | | | | | | | | | | | | | |
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 1.80 | 2.30 | 3.30 | 3.40 | 3.40 | 3.40 | 4.50 | 4.50 | 4.50 | 7.00 | 7.90 | 8.40 | 8.80 | 12.80 | 13.10 | 13.10 | 13.10 | 13.10 | 13.10 |
| 1 | 1.00 | 1.60 | 1.80 | 2.60 | 3.10 | 3.30 | 3.30 | 4.10 | 4.80 | 6.30 | 7.40 | 9.10 | 9.60 | 11.20 | 12.50 | 12.50 | 12.50 | 12.50 | 12.50 | 12.50 |
| 2 | 1.00 | 1.50 | 2.30 | 2.70 | 3.00 | 3.00 | 3.00 | 4.80 | 5.20 | 5.20 | 6.50 | 7.10 | 8.30 | 9.40 | 10.50 | 11.80 | 11.80 | 11.80 | 11.80 | 11.80 |
| *tecator* | *H* | | | | | | | | | | | | | | | | | | | |
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 1.10 | 1.50 | 1.50 | 1.50 | 1.50 | 1.50 | 1.50 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 | 2.20 |
| 1 | 1.00 | 1.40 | 1.60 | 1.90 | 2.60 | 2.60 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 | 3.20 |
| 2 | 1.00 | 1.50 | 1.70 | 2.00 | 2.00 | 2.40 | 2.40 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 | 2.90 |
| *trigonometric* | *H* | | | | | | | | | | | | | | | | | | | |
| *d* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 1.00 | 2.00 | 2.40 | 2.50 | 2.50 | 2.50 | 3.00 | 4.10 | 4.10 | 4.10 | 4.10 | 4.90 | 4.90 | 4.90 | 4.90 | 4.90 | 4.90 | 5.50 | 5.50 | 5.50 |
| 1 | 1.00 | 2.00 | 2.30 | 2.40 | 2.60 | 2.60 | 2.90 | 3.50 | 3.50 | 3.50 | 3.50 | 3.50 | 4.10 | 4.10 | 4.10 | 4.10 | 4.10 | 4.10 | 4.10 | 4.10 |
| 2 | 1.00 | 2.00 | 2.20 | 2.40 | 2.40 | 2.40 | 2.40 | 3.00 | 3.00 | 3.80 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 | 4.70 |

### 4.2.3. Results on the optimal choice of the time instants, H

In order to obtain the best number of time instants, $H$, we performed cross-validation on the validation sample $s_3$, as is detailed in Section 3.3. Thanks to the nested structure of our algorithm, we are able to build a trajectory, from $h = 1$ to $h = H$, in which the evolution of the optimal number of time instants can be observed. Particularly, Table 4 shows the average optimal number of time instants over all the folds in the univariate and multivariate databases. Moreover, in Figs. 5 and 6 the resulting boxplots are depicted. In the $x-$axis, the maximum number of time instants considered when running our heuristic is given, whereas the $y-$axis indicates the optimal number of time instants obtained across the different runs. Boxplots in red, blue and green show the results when the information of the derivative $d = 0$, $d = 1$ or $d = 2$ is used, respectively.

We can observe that, although the experiments are run until $H = 20$, the optimal number of time instants to be selected is lower in almost all databases. Indeed, most of the datasets need between 1 and 8 time instants. It implies that data information is summarized on a small finite set of time points, which may yield good interpretation results.

## 5. Conclusions and extensions

We have proposed in this paper a new approach able to optimally select the most informative time instants in multivariate functional data in order to get good classification rates. Furthermore, our methodology, by its nature, allows the easy usage of high-order information, e.g. monotonicity, or convexity by means of the derivatives. The numerical experience here presented has shown that the information provided by the derivatives has valuable consequences in the classification performance, yielding competitive results when compared against the state-of-the-art in the literature. We have worked under the assumption that time is a continuous parameter, and continuous optimization tools are then used to optimize the parameters.

The nested structure of the problem improves the current methodology by using the optimal solutions obtained in simpler models as starting solutions in more complex models.

In our analysis, for the sake of simplicity, we have considered the Pearson correlation as the performance measure to be optimized. Nevertheless, other measures such as the Mutual Information Criterion [22], the Fisher-Correlation Criteria, [16], the distance covariance [4,41,44], or the distance correlation in [44] can be used. In this paper, we restricted ourselves to the multivariate functional data case. The problem of time instants selection in multivariate hybrid functional data [26] is also worth being analyzed. Possible extension of this work to the clustering context deserve further study. The extension to the regression area is being analyzed in [9]. Here, we have just employed the information provided by the first and second derivatives. Thanks to kernel definition, it is very easy to extend our proposal, in order to include the derivatives of order equal or greater than three.

## Acknowledgments

Research partially supported by research grants MTM2015-65915-R (Ministerio de Ciencia e Innovación, Spain), P11-FQM-7603, FQM329 (Junta de Andalucía Spain), FPU (Ministerio de Educación, Cultura y Deporte), all with EU ERDF funds, as well as FBBVA-COSECLA. This support is gratefully acknowledged. We also thank the team of the Scientific Computing Center of Andalucía (CICA) for the computing services they provided.

## References

[1] A. Baíllo, A. Cuevas, R. Fraiman, Classification methods for functional data, in: The Oxford Handbook of Functional Data Analysis, Oxford University Press, 2011, pp. 259–297.

[2] S. Benítez-Peña, R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, Cost-sensitive feature selection for support vector machines, Comput. Oper. Res. (2018), doi:10.1016/j.cor.2018.03.005.

[3] J. Berrendero, A. Justel, M. Svarc, Principal components for multivariate functional data, Comput. Stat. Data Anal. 55 (9) (2011) 2619–2634.

[4] J.R. Berrendero, A. Cuevas, J.L. Torrecilla, Variable selection in functional data classification: a maxima-hunting proposal, Stat. Sin. 26 (2) (2016) 619–638.

[5] J.R. Berrendero, A. [Cuevas], & J.L. [Torrecilla] (2016). Variable selection in functional data classification: a maxima-hunting proposal. Statistica Sinica, 26, 619–638.

[6] J.R. Berrendero, A. Cuevas, J.L. Torrecilla, On the use of reproducing kernel hilbert spaces in functional classification, J. Am. Stat. Assoc. (2017), doi:10.1080/01621459.2017.1320287.

[7] R. Blanquero, E. Carrizosa, O. Chis, N. Esteban, A. Jiménez-Cordero, J.F. Rodríguez, M.R. Sillero-Denamiel, On extreme concentrations in chemical reaction networks with incomplete measurements, Ind. Eng. Chem. Res. 55 (2016) 11417–11430.

[8] R. Blanquero, E. Carrizosa, A. Jiménez-Cordero, B. Martín-Barragán, Functional-bandwidth kernel for support vector machine with functional data: an alternating optimization algorithm, Eur. J. Oper. Res. (2018), doi:10.1016/j.ejor.2018.11.024.

[9] R. Blanquero, E. Carrizosa, A. Jiménez-Cordero, B. Martín-Barragán, Variable selection with support vector regression for multivariate functional data, Technical Report, University of Edinburgh - Universidad de Sevilla, 2018. Available at https://www.researchgate.net/publication/327552293_Variable_Selection_with_Support_Vector_Regression_for_Multivariate_Functional_Data.

[10] R. Blanquero, E. Carrizosa, A. Jiménez-Cordero, J.F. Rodríguez, A global optimization method for model selection in chemical reactions networks, Comput. Chem. Eng. 93 (2016) 52–62.

[11] E. Carrizosa, B. Martín-Barragán, D. Romero Morales, A nested heuristic for parameter tuning in support vector machines, Comput. Oper. Res. 43 (2014) 328–334.

[12] B. Colson, P. Marcotte, G. Savard, An overview of bilevel optimization, Ann. Oper. Res. 153 (1) (2007) 235–256.

[13] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[14] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
[15] C. De Boor, A Practical Guide to Splines, Applied Mathematical Sciences, 27, Springer-Verlag New York, 1978.
[16] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinform. Comput. Biol. 3 (2005) 185–205.
[17] F. Ferraty, P. Hall, P. Vieu, Most-predictive design points for functional data predictors, Biometrika 97 (4) (2010) 807–824.
[18] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer Science & Business Media, 2006.
[19] M.C. Ferris, T.S. Munson, Semismooth support vector machines, Math. Program 101 (1) (2004) 185–204.
[20] J. Friedman, T. Hastie, R. Tibshirani, Datasets for *The Elements of Statistical Learning*, 2001, (https://web.stanford.edu/~hastie/ElemStatLearn/data.html).
[21] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer Series in Statistics, 1, Springer, Berlin, 2001.
[22] V. Gómez-Verdejo, M. Verleysen, J. Fleury, Information-theoretic feature selection for functional data classification, Neurocomputing 72 (16) (2009) 3580–3589. Financial Engineering Computational and Ambient Intelligence (IWANN 2007)
[23] F. Ieva, A.M. Paganoni, D. Pigoli, V. Vitelli, Multivariate functional clustering for the morphological analysis of electrocardiograph curves, J. R. Stat. Soc. Ser. C (Appl. Stat.) 62 (3) (2013) 401–418.
[24] J. Jacques, C. Preda, Model-based clustering for multivariate functional data, Comput. Stat. Data Anal. 71 (2014) 92–106.
[25] G.M. James, J. Wang, J. Zhu, Functional linear regression that's interpretable, Ann. Stat. 37 (2009) 2083–2108.
[26] A. Jiménez-Cordero, S. Maldonado, Automatic feature scaling and selection for support vector machine classification with functional data, Technical Report, Universidad de los Andes - Universidad de Sevilla, 2018. Available at https://www.researchgate.net/publication/323428879_Automatic_Feature_Scaling_and_Selection_for_Support_Vector_Machine_Classification_with_Functional_Data.
[27] S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in: Proceedings of the Science and Information Conference, 2014, pp. 372–378.
[28] Y. Li, M. Dong, J. Hua, Localized feature selection for clustering, Pattern Recognit. Lett. 29 (1) (2008) 10–18.
[29] M.A. Lindquist, I.W. McKeague, Logistic regression with Brownian-like predictors, J. Am. Stat. Assoc. 104 (488) (2009) 1575–1585.
[30] N. Locantore, J.S. Marron, D.G. Simpson, N. Tripoli, J.T. Zhang, K.L. Cohen, G. Boente, R. Fraiman, B. Brumback, C. Croux, J. Fan, A. Kneip, J.I. Marden, D. Peña, J. Prieto, J.O. Ramsay, M.J. Valderrama, A.M. Aguilera, Robust principal component analysis for functional data, Test 8 (1) (1999) 1–73.
[31] J. López, S. Maldonado, M. Carrasco, Double regularization methods for robust feature selection and SVM classification via dc programming, Inf. Sci. (Ny) 429 (2018) 377–389.
[32] B. Martín-Barragán, R. Lillo, J. Romo, Interpretable support vector machines for functional data, Eur. J. Oper. Res. 232 (1) (2014) 146–155.
[33] T. Mehmood, K.H. Liland, L. Snipen, S. Sæbø, A review of variable selection methods in partial least squares regression, Chemom. Intell. Lab. Syst. 118 (2012) 62–69.
[34] A. Muñoz, J. González, Representing functional data using support vector machines, Pattern Recognit. Lett. 31 (6) (2010) 511–516.
[35] R.E. Naftchali, M.S. Abadeh, A multi-layered incremental feature selection algorithm for adjuvant chemotherapy effectiveness/futileness assessment in non-small cell lung cancer, Biocybern. Biomed. Eng. 37 (3) (2017) 477–488.
[36] H. Peng, Y. Fan, Feature selection by optimizing a lower bound of conditional mutual information, Inf. Sci. (Ny) 418–419 (2017) 652–667.
[37] C. Preda, G. Saporta, C. Lévéder, PLS classification of functional data, Comput. Stat. 22 (2) (2007) 223–235.
[38] J.O. Ramsay, B.W. Silverman, Applied Functional Data Analysis: Methods and Case Studies, Springer Series in Statistics, 77, Springer-Verlag, 2002.
[39] J.O. Ramsay, B.W. Silverman, Functional Data Analysis, Springer Series in Statistics, Second ed., Springer-Verlag, 2005.
[40] F. Rossi, N. Villa, Support vector machine for functional data classification, Neurocomputing 69 (7) (2006) 730–742.
[41] G.J. Székely, M.L. Rizzo, N.K. Bakirov, et al., Measuring and testing dependence by correlation of distances, Ann. Stat. 35 (6) (2007) 2769–2794.
[42] R. Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2017. https://www.R-project.org/.
[43] J.L. Torrecilla, A. Suárez, Feature selection in functional data classification with recursive maxima hunting, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 4835–4843.
[44] J.L. Torrecilla Noguerales, On the theory and practice of variable selection for functional data, Universidad Autónoma de Madrid, 2015 Ph.D. thesis.
[45] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[46] H. Wang, M. Yao, Fault detection of batch processes based on multivariate functional kernel principal component analysis, Chemom. Intell. Lab. Syst. 149 (2015) 78–89.
[47] X. Wang, S. Ray, B.K. Mallick, Bayesian curve classification using wavelets, J. Am. Stat. Assoc. 102 (479) (2007) 962–973.
[48] X. Zhang, B.U. Park, J.-L. Wang, Time-varying additive models for longitudinal data, J. Am. Stat. Assoc. 108 (503) (2013) 983–998.
[49] Y. Zhao, R.T. Ogden, P.T. Reiss, Wavelet-based lasso in functional linear regression, J. Comput. Gr. Stat. 21 (3) (2012) 600–617.
[50] A. Zibakhsh, M.S. Abadeh, Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function, Eng. Appl. Artif. Intell. 26 (4) (2013) 1274–1281.