European Journal of Operational Research 275 (2019) 195–207

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

Functional-bandwidth kernel for Support Vector Machine with Functional Data: An alternating optimization algorithm

R. Blanquero[a], E. Carrizosa[a], A. Jiménez-Cordero[a,*], B. Martín-Barragán[b]

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

ARTICLE INFO

Article history:
Received 8 May 2018
Accepted 8 November 2018
Available online 24 November 2018

Keywords:
Data mining
Functional Data classification
Parameter tuning
SVM
Functional bandwidth

ABSTRACT

Functional Data Analysis (FDA) is devoted to the study of data which are functions. Support Vector Machine (SVM) is a benchmark tool for classification, in particular, of functional data. SVM is frequently used with a kernel (e.g.: Gaussian) which involves a scalar bandwidth parameter. In this paper, we propose to use kernels with functional bandwidths. In this way, accuracy may be improved, and the time intervals critical for classification are identified. Tuning the functional parameters of the new kernel is a challenging task expressed as a continuous optimization problem, solved by means of a heuristic. Our experiments with benchmark data sets show the advantages of using functional parameters and the effectiveness of our approach.

1. Introduction

Functional Data Analysis (FDA) has received considerable attention from researchers, (Ferraty & Vieu, 2006; Ramsay & Silverman, 2002, 2005; Wang, Chiou, & Müller, 2016) and practitioners in many different fields, such as spectrometry, meteorology, (Martín-Barragán, Lillo, & Romo, 2014), client segmentation, (Laukaitis & Račkauskas, 2005), speech recognition, (Rossi & Villa, 2006), or physical, (Muñoz & González, 2010; Tuddenham & Snyder, 1954), and chemical processes, (Blanquero et al., 2016a; Blanquero, Carrizosa, Jiménez-Cordero, & Rodríguez, 2016b).

FDA can be considered as a generalization of the standard multivariate analysis to address problems in which data have an infinite-dimensional nature. The direct application of classic methods of multivariate analysis on infinite-dimensional data may have dramatic consequences in the obtained results. The curse of dimensionality is a clear example of this situation. Indeed, although theoretically data are described as functions, in practice functional data are represented by high dimensional vectors, yielding problems in which the number of observations is lower than the number of features and which cannot be handled by standard multivariate analysis tools. Furthermore, it is worthwhile to mention that the methodologies used for multivariate vectors do not exploit the functional behavior of the data since the high correlations among the different coordinates are not taken into account.

In this work, we focus on a challenging problem in FDA: functional binary classification, i.e., how to classify functional data into two predefined classes using the information provided by a training sample (Baíllo, Cuevas, & Cuesta-Albertos, 2011; Biau, Bunea, & Wegkamp, 2005; Cuevas, Febrero, & Fraiman, 2007; Ferraty & Vieu, 2006; Preda, Saporta, & Lévéder, 2007). Support Vector Machine (SVM) (Carrizosa & Romero Morales, 2013; Cauwenberghs & Poggio, 2001; Cortes & Vapnik, 1995; Cristianini & Shawe-Taylor, 2000; Lessmann & Voß, 2009; Maldonado, Pérez, & Bravo, 2017; Maldonado, Weber, & Basak, 2011; Suykens & Vandewalle, 1999; Wang, Zheng, Yoon, & Ko, 2018) is one of the most used tools in multivariate classification, and it has also been widely applied for functional data. See Blanquero, Carrizosa, Jiménez-Cordero, and Martín-Barragán (2017), Jiménez-Cordero and Maldonado (2018), Martín-Barragán et al. (2014), Muñoz and González (2010), Rossi and Villa (2006), and Rossi and Villa (2008) among others.

As stated before, solving functional data problems, and more specifically, the functional classification problem, implies the use of specific techniques that take advantage of the intrinsic functional nature of the data.

For SVM, (Rossi & Villa, 2006) exploits the functional behavior of the data by adapting the classical kernels to functional kernels through the so-called transformation-based and projection-based kernels. Nevertheless, the whole range of the data is weighted with a single scalar bandwidth.

* 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.ejor.2018.11.024
0377-2217/

The functional nature of the data is taken into account in Kästner, Hammer, Biehl, and Villmann (2012), generalizing the work done in the multivariate case in Hammer and Villmann (2002); Sato and Yamada (1996). Data are classified according to a dissimilarity measure with a functional weight. Such a functional weight is represented in terms of simple basis functions whose parameters are sought via stochastic gradient.

To the best of our knowledge, no strategy has been presented in the literature using a supervised tool, e.g., SVM, in which different ranges in the domain of the functions are optimally selected by means of a functional weight in the kernel. Therefore, one of the main contributions of this paper is to define a new functional kernel. Such kernel has a functional bandwidth that optimally weighs the different values of the domain of the function. Similar ideas have been used in references such as Bugeau and Pérez (2007), Chen, Wynne, Goulding, and Sandoz (2000), Duong, Cowling, Koch, and Wand (2008), and Sain (2002) for kernel density estimation, and in Cai, Fan, and Yao (2000) and Wu, Chiang, and Hoover (1998) for functional regression.

We propose to embed the new functional kernel into an SVM algorithm. Both the kernel and the SVM parameters are tuned with a surrogate of the accuracy, namely, the correlation between the actual class and the SVM score. See also Berrendero, Cuevas, and Torrecilla (2016), Székely, Rizzo, Bakirov et al. (2007), and Torrecilla Noguerales (2015) for more details on the use of surrogate measures for the accuracy. Such parameter tuning yields a continuous optimization problem, allowing us to use gradient methods, known to be more efficient than the optimization methods available for piecewise constant performance measures, such as the misclassification rate. Moreover, the proposed method is enhanced by defining a hierarchy of kernel bandwidths models of increasing complexity, inspired by the nested model previously proposed for Multiple Kernel Learning in Carrizosa, Martín-Barragán, and Romero Morales (2014). Using this hierarchy provides wide flexibility since complex parameterizations of the functional bandwidth can be efficiently optimized from more simple ones.

The remainder of the paper is structured as follows. In Section Section 2 we present the SVM classification model for functional data. Section 3 describes the optimization method used to tune the bandwidth parameters. We focus on the alternating procedure proposed to this purpose, and on the structure of the hierarchy of kernels. Section 4 is devoted to the numerical experiments, showing that our approach outperforms the method in which one single scalar parameter bandwidth is chosen. Finally, some conclusions and extensions are described in Section 5.

2. Functional bandwidth

In this section, we formulate the SVM problem for functional data classification. See Cristianini and Shawe-Taylor (2000) for a broader and more comprehensive presentation of SVM. We have a sample s of observations; each observation $i \in s$ has associated a pair (X_i, Y_i), where each $X_i : [0, T] \to \mathbb{R}$ belongs to the set $\mathcal{X}$ of Riemann integrable functions in the time interval $[0, T]$. Furthermore, $Y_i \in \{-1, +1\}$ denotes the class label for the observation i. Our goal is to find a classification rule to infer the class Y of a new functional observation $X \in \mathcal{X}$.

The well-known technique SVM considers a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, (Cristianini & Shawe-Taylor, 2000; Rossi & Villa, 2006, 2008), and builds, from a sample s, nonlinear classifiers by means of a score $\hat{Y}(X)$ of the form:

$$\hat{Y}(X) = \sum_{i \in s} \alpha_i Y_i K(X, X_i), \quad X \in \mathcal{X}, \tag{1}$$

yielding the following classification rule: a functional observation $X \in \mathcal{X}$ is assigned to class $+1$ if and only if $\hat{Y}(X) > \beta$, where β is a given threshold value. Here the values α_i, $i \in s$, are obtained as the optimal solution of the following optimization problem:

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

for a scalar regularization parameter C to be tuned, usually by k-fold cross-validation with a grid search on a sufficiently large interval.

Many types of kernels for data in $\mathbb{R}^d$ are proposed in the literature, e.g., the linear kernel, the polynomial kernel, or the Gaussian (RBF) kernel, given by:

$$K(X_i, X_j) = \exp\left(- \sum_{t=1}^{d} (X_{it} - X_{jt})^2 \omega \right), \quad X_i, X_j \in \mathbb{R}^d \tag{3}$$

where ω is a scalar bandwidth to be tuned, (Carrizosa et al., 2014; Carrizosa & Romero Morales, 2013; Cristianini & Shawe-Taylor, 2000; Hofmann, Schölkopf, & Smola, 2008; Keerthi & Lin, 2003). In this paper, for simplicity, we only focus on the Gaussian kernel, one of the most used and effective kernels, which will be used in what follows.

The expression (3) of the Gaussian kernel for data in $\mathbb{R}^d$ has been generalized to a Gaussian kernel for functional data, e.g., Kadri, Duflos, Preux, Canu, and Davy (2010) and Wang and Yao (2015). Nevertheless, in these papers, the associated bandwidth is always considered to be a scalar value. In our proposal we extend the fixed scalar bandwidth parameter ω in an RBF kernel to a functional bandwidth, $\omega(t)$, that varies along the range of the functional data, (4):

$$K(X_i, X_j) = \exp\left(- \int_0^T (X_i(t) - X_j(t))^2 \omega(t) dt \right) \tag{4}$$

Throughout this paper, we assume that ω in (4) is a non-negative Riemann integrable function in $[0, T]$, and thus K is well-defined.

It is worth mentioning that the simplest extension from the kernel with vector data (3) to the kernel with functional data (4) would be to consider $\omega(t)$ as a constant function, as in Kadri et al. (2010) and Wang and Yao (2015). Nevertheless, the main contribution of this paper is to consider such bandwidth as a function which adapts to the structure and shape of the data and may lead to better insight and classification rates. More specifically, making ω depend on t allows us to identify those subintervals in $[0, T]$ which are critical for classification, namely, those for which $\omega(t)$ takes highest values.

Example 2.1. As an illustration, let us study the *regions* data set (Martín-Barragán et al., 2014), in which the daily temperature has been measured along a year in each of 35 Canadian weather stations. Two groups can be distinguished: Atlantic climate (label -1), with 15 records, versus the rest of climates (label 1), with 20 records. Our objective is to discriminate between both classes. Fig. 1 depicts the 15 curves in the interval [1, 365] corresponding to the Atlantic climate, in solid black line, and the 20 curves corresponding to the rest of climates, in dashed red line, with the data measured every single day. This is, by nature, a Functional Data classification problem. However, it may be considered as a classic classification problem with 15+20 records in $\mathbb{R}^d$, $d = 365$ (the number of time instants at which the temperature has been actually recorded), and thus one can apply the classic SVM in the form (3) for some ω to be tuned. Observe that this model is the same as model (4) with

$$\omega(t) = \omega, \quad \forall t \in [0, T] \quad with \ T = 365, \tag{5}$$

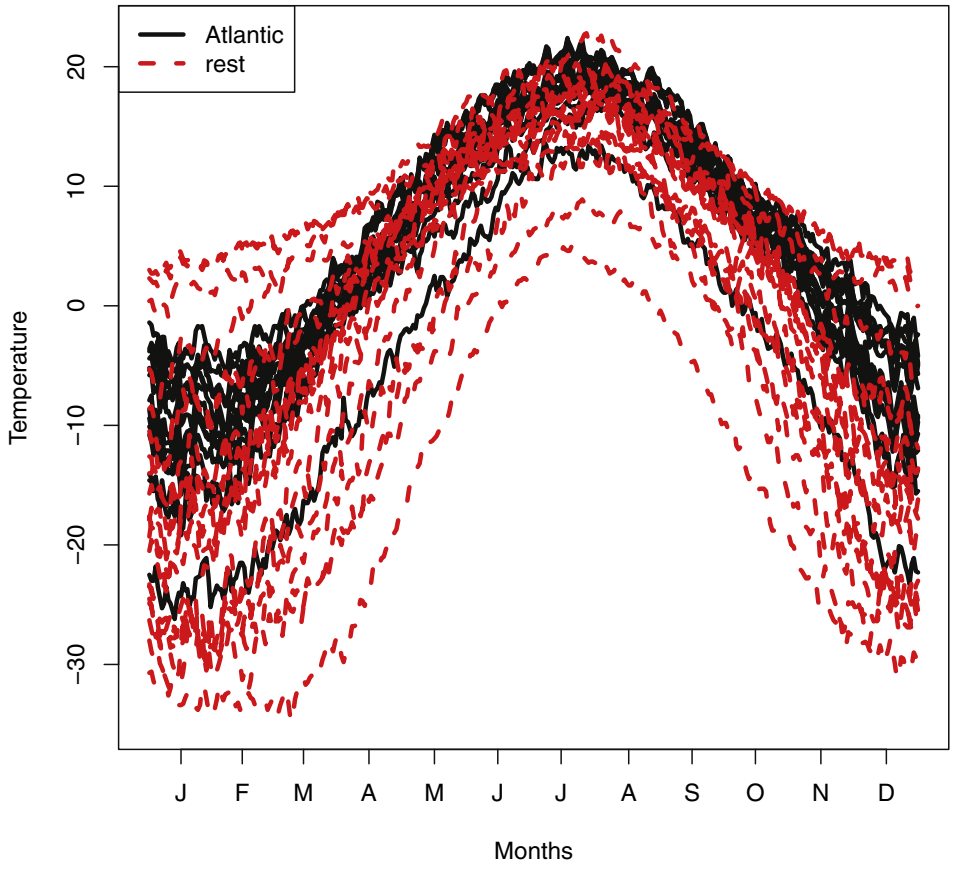


Fig. 1. *regions* data set.

Table 1
Confusion matrix with ω as in (5).

	Label −1	Label 1
Label −1	51.42%	5.71%
Label 1	11.42%	31.42%

Table 2
Confusion matrix with ω as in (6).

	Label −1	Label 1
Label −1	54.28%	2.85%
Label 1	8.57%	34.28%

and the integral evaluated numerically in the grid of time instants where the temperature is measured. Using SVM with a constant $\omega(t)$ as in (5) leads to a classifier with the out-of-sample confusion matrix shown in Table 1.

Now, let us consider the very same RBF model with a functional bandwidth $\omega(t)$ of the form

$$\omega(t) = \begin{cases} \omega_1, & if\ 0 \le t \le \tau_1 \\ \omega_2, & if\ \tau_1 < t \le 365, \end{cases} \tag{6}$$

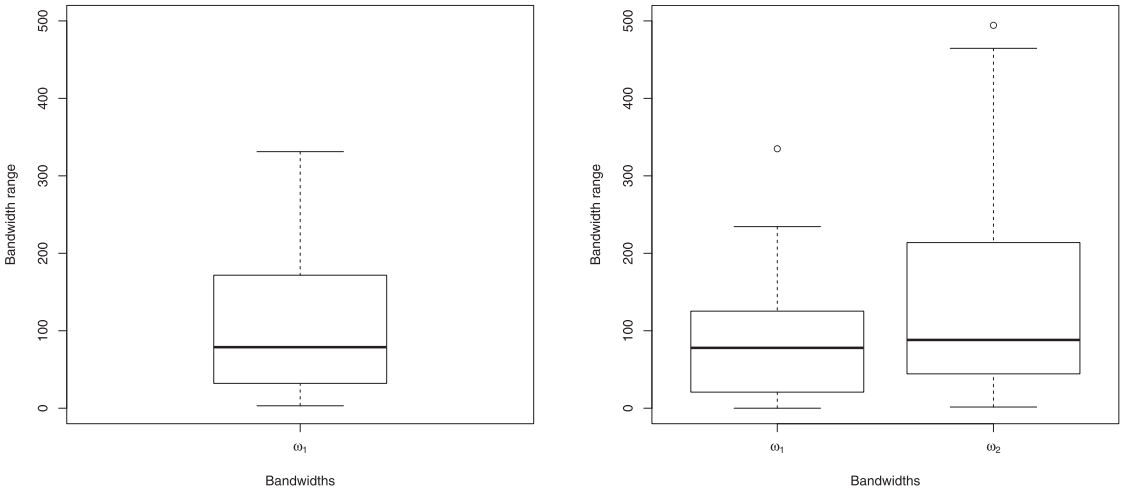
where ω_1, ω_2, τ_1 are parameters to be tuned using the techniques described in this paper. In other words, with the bandwidth in (6) we split the interval $[0, T]$ into two pieces, giving different weights to each time interval. The SVM classifier obtained this way leads to the out-of-sample confusion matrix in Table 2.

Comparing Tables 1 and 2 we can see that the traditional SVM yields an accuracy of 82.84%. On the other hand, our SVM with the very same RBF kernel but using a functional parameter of the form (6) yields an accuracy of 88.56%, instead.

Regarding the interpretability of the results, Figs. 2 and 3 show the boxplots of the values of the bandwidth ω as in (5), and the values of ω_1, ω_2 and τ_1, as in (6). The single-bandwidth approach gives the same importance to all the months of the year with the majority of the bandwidth values between 50 and 150. In contrast, our functional-bandwidth methodology with two different pieces proposed to divide the whole year into two parts, before and after summer (months of June and July), see Fig. 3. Moreover, according to the values of ω_1 and ω_2, in order to get good classification predictions, we should focus on the second half year and give more importance to the second part, i.e., the autumn and first months of winter, which coincides to the time instants when the temperature begins to decrease.

The previous illustrative example demonstrates that even a simple functional bandwidth such as (6) may yield improvements in accuracy. Such improvement is a consequence of the adequate choice of the parameter τ_1, which combined with good values of ω_1 and ω_2 allow us to identify the suitable intervals for classification. The functional bandwidth parameter $\omega(t)$ gives more flexibility, which should result in greater precision. For instance, it may be chosen in the class of piecewise constant non-negative functions in $[0, T]$ with H pieces, i.e., one can naturally assume that $\omega(t)$ has the form (7)

$$\omega(t) = \begin{cases} \omega_1, & if\ 0 \le t \le \tau_1 \\ \omega_2, & if\ \tau_1 < t \le \tau_2 \\ \cdots & \\ \omega_h, & if\ \tau_{h-1} < t \le \tau_h \\ \cdots & \\ \omega_H, & if\ \tau_{H-1} < t \le T \end{cases} \tag{7}$$



(a) Bandwidth value with ω as in (5)

(b) Bandwidth value with ω as in (6)

Fig. 2. (a) and (b) Show the bandwidth values for the *regions* data set when ω has the form of (5) and (6), respectively.

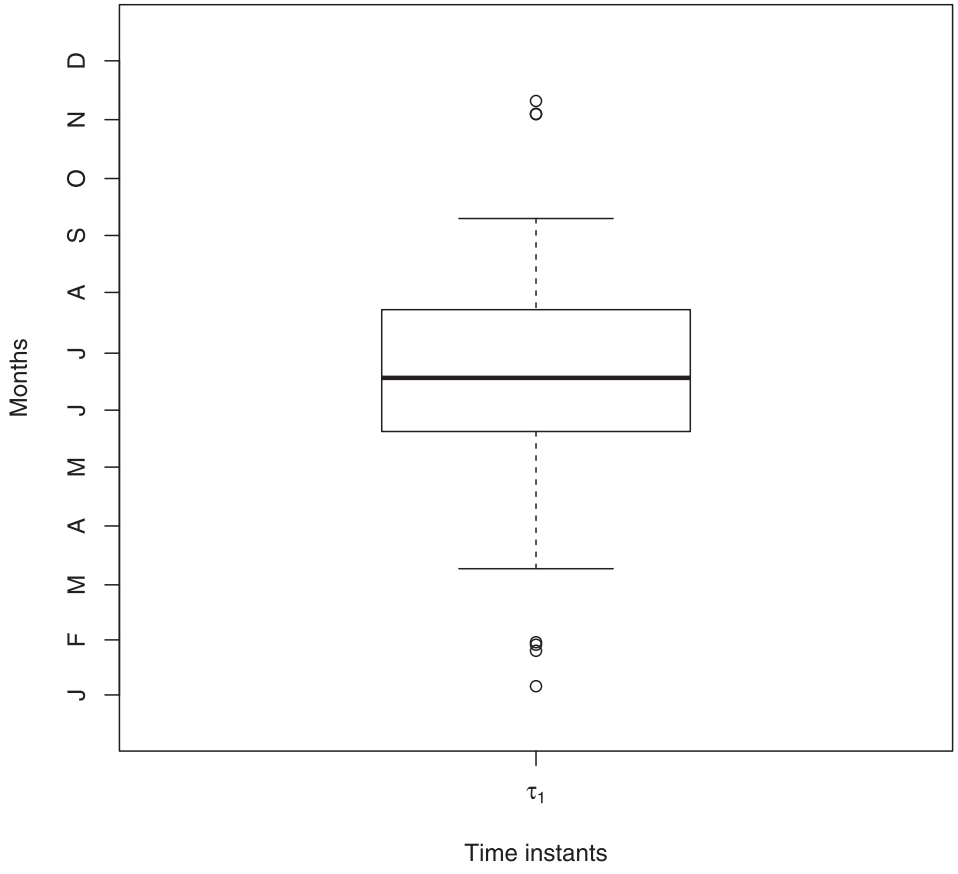


Fig. 3. Time instant results for the *regions* data set with ω as in (6).

where $\omega_1, \ldots, \omega_H \geq 0$ and $0 \leq \tau_1 \leq \cdots \leq \tau_{H-1} \leq T$ are parameters to be tuned. Instead of piecewise constant functions, one could consider $\omega(t)$ belonging to the class of polynomials of degree H which are non-negative in $[0, T]$, the class of piecewise polynomial functions non-negative in $[0, T]$, or the non-negative splines, (De Boor, 1978; Friedman, Hastie, & Tibshirani, 2001b).

The use of functional parameters in the kernel may lead to significant improvements in the accuracy, as demonstrated in our numerical experiments. The price to pay for obtaining such gains in the accuracy is the fact that tuning the functional parameters calls for using more sophisticated optimization procedures. In Section 3 we detail how the underlying optimization problem for tuning $\omega(t)$ is solved.

3. Optimal selection of the functional bandwidth

In this Section, a detailed study of the mathematical formulation of the (functional) parameter tuning in SVM classification is presented. Section 3.1 explains how to formulate the optimization problem involved and how to solve it. In Section 3.2 a nested

heuristic to address the tuning problem more efficiently is described. In this way, we exploit the fact that the bandwidths considered are elements of a nested family of kernels. Section 3.3 details how to choose the number of pieces, H, of the functional bandwidth.

3.1. Problem formulation and optimization

Parameter tuning in the classification of functional data with SVM implies the optimal choice of two very different elements: the scalar regularization parameter C in (2), and the kernel K in (4) through $\omega(t)$. The problem of finding the best function $\omega(t)$ in (4), is not tractable as a rule in its full generality. Hence, we restrict our attention to certain classes of functions parameterized by a vector θ belonging to a certain set Θ, i.e., ω is expressed as $\omega(t, \theta)$, and the choice of the function ω is equivalent to choosing the parameters θ.

Example 3.1. For the bandwidth given in (7), one would have that $\theta = (\omega_1, \ldots, \omega_H, \tau_1, \ldots, \tau_{H-1})$, and $\Theta = \{(\omega_1, \ldots, \omega_H, \tau_1, \ldots, \tau_{H-1}) : \omega_h \geq 0,\ \forall h,\ \tau_h \in [0, T],$ $h = 1, \ldots, H-1,\ \tau_1 \leq \cdots, \leq \tau_{H-1}\}$. For convenience, we consider $\tau_0 = 0$ and $\tau_H = T$.

In principle, in order to find the optimal values of the parameters, C, and θ, a strategy based on a grid search on both parameters could be applied. Given a set of predefined pairs of values (C, θ), one first solves (2) to obtain the coefficients α of the score function (1), and then the corresponding accuracy associated to that pair is computed. However, this approach may become too time consuming, and thus a more sophisticated heuristic is proposed in these lines. We propose to follow the standard grid approach to optimize C. Nevertheless, when seeking the parameters θ and α we propose to solve a bilevel problem where some measure of the quality of θ is maximized for the α provided by the SVM classifier, i.e., for the α solving (2).

Many criteria can be chosen to guide the choice of parameter θ. One may, for instance, minimize the misclassification rate, which is the default approach for tuning the parameter C. However, the misclassification rate has a discrete nature that would prevent us from using continuous optimization techniques, and, in particular, from gradient-based methods. Instead, we propose to maximize the Pearson correlation, R, between the class label Y_i of the functional data X_i and the score, $\hat{Y}(X_i, \theta, \alpha)$ in (1), where all the variables, including the time instants, are treated as continuous variables. Other references in the literature, such as Blanquero et al. (2017) and Jiménez-Cordero and Maldonado (2018), have previously used with excellent results the Pearson correlation coefficient. Despite the fact that, when using the Pearson correlation coefficient as a surrogate of accuracy, a linear relationship between the binary label, $Y \in \{-1, 1\}$, and the real-valued score, $\hat{Y} \in \mathbb{R}$, is implicitly assumed, this coefficient is very fast to compute, and even more important, it also allows us to use gradient-based methodologies since its optimization amounts to solving a continuous optimization problem.

It is very well known that building a classifier and evaluating its performance over the same data set leads to overfitting. In such a case, the model fits the data set too well but performs poorly in unseen data. On top of that, the classifier can depend on parameters that must be tuned, usually done by performing a grid search in a suitable range of values. The usual way to avoid overfitting in this general situation is to split the data set, perhaps within a k-fold cross-validation framework, in three parts, the so-called training, validation, and test samples. For a given choice of the parameters, the first two ones are used to build the model and estimate its performance, respectively; once the best parameters have been chosen, the final model is tested on the last sample. In our case, we take this idea further by creating four independent samples, due to the structure of our resolution method. First, the data set is divided into k folds. Second, $k-1$ folds are again split into three samples named s_1, s_2, and s_3, while the remaining fold is denoted by s_4. Samples s_1 and s_2 play the role of training samples, whereas s_3 and s_4 form the validation and testing sets, respectively, as will be detailed next.

The first independent sample s_1 is employed for the resolution of Problem (2), that is the classic SVM formulation, to obtain a classification rule by means of α, for fixed parameters θ and C. The second independent sample s_2 is used to measure the quality of parameters θ, i.e., it is used to calculate $R((Y_i, \hat{Y}(X_i, \theta, \alpha))_{i \in s_2})$, the correlation between the class labels and the scores. To find the regularization parameter C, we measure the accuracy in the sample s_3 for all the different possible values of C in the grid, and we keep the C providing the largest accuracy. Finally, the accuracy in the independent sample s_4 is measured.

After all these considerations, for fixed C, the bilevel problem can be expressed as:

$$\begin{cases} \max\limits_{\theta,\alpha} & R((Y_i, \hat{Y}(X_i, \theta, \alpha))_{i \in s_2}) \\ s.t. & \alpha \text{ solves (2) in } s_1 \\ & \theta \in \Theta \end{cases} \tag{8}$$

Note that we have emphasized the dependence of the score $\hat{Y}$ on θ and α by including them in the notation. In the cases where the values of the parameters in θ, or the classification coefficients α are clear enough, we will omit them for the sake of simplicity.

Problem (8) is a nonlinear bilevel optimization problem, which can be handled with off-the-shelf strategies, as those described in Colson, Marcotte, and Savard (2007). These techniques are, however, rather expensive. Recall that (8) is only a surrogate of our real problem. Hence, instead of the above-mentioned standard methodologies, we next propose an alternating approach for which only a few iterations will be carried out. Firstly, in the first step of our alternating approach, for fixed parameters θ and C, a classification rule is obtained by means of α solving Problem (2), that is, the classic SVM formulation. Problem (2) is a concave quadratic maximization problem, which can be solved by standard local search optimizers or specific routines, as in Ferris and Munson (2004); Richtárik and Takáč (2016). Secondly, in the second step, for fixed α and C, θ is chosen by solving:

$$\max_{\theta \in \Theta} R((Y_i, \hat{Y}(X_i, \theta))_{i \in s_2}) \tag{9}$$

Problem (9) is a continuous optimization problem which is solved by using standard local search techniques within a multi-start strategy. The alternating procedure will alternate these two steps until some stopping criterion is met. Suitable values for θ and α will be obtained by this procedure for a specific value of the regularization parameter C.

The value of C will be chosen by a grid search, as commonly done in standard SVM. This means that, for every value of C in a given grid, we measure the accuracy in s_3 of the classification rule obtained with the best θ and α found as solutions of Problem (9). The C with the largest accuracy in s_3 will be chosen. Finally, we estimate the correct classification rate using the fourth independent sample, s_4.

A pseudocode of the heuristic is outlined in Algorithm 1, and in Section 3.2, we detail an extension to more complex models by means of a nested heuristic, described above.

3.2. Optimization enhancement. A nested optimization

When the dimension of θ is high, the approach described in Section 3.1 may be time-consuming. The main reason is that, on top of the grid search needed for C, Problem (9) may have many

Algorithm 1 Heuristic for parameter tuning.

Input: H
• Randomly split the sample s into s_1, s_2, s_3 and s_4.
for C in the grid **do**
Alternating Procedure
repeat
1. Fixed θ, calculate the parameters α of the SVM classifier by solving Problem (2) in s_1.
2. Fixed α, calculate θ by solving Problem (9) in s_2.

until stopping criteria
• Evaluate the accuracy in the sample s_3 with C fixed.
end for
• The optimal value of C is the one with the best accuracy in s_3. The optimal values of α and θ are the ones associated to the optimal parameter C.
Output: optimal parameters C and θ, optimal classification coefficients α, and the corresponding accuracy estimated from s_4.

local minima, and therefore multiple local searches are required to find a good solution. The success of the method would be improved if, instead of a random multi-start, a more intelligent search strategy were possible. This is the case, for instance, for models of the bandwidth parameter $\omega(t, \theta)$ that can be plugged into a sequence of models of increasing complexity. Thus the optimal solution obtained in the simple model can be used as a starting solution in the following more complex case.

The above-explained methodology can be easily embedded in a nested heuristic for SVM parameter tuning, Carrizosa et al. (2014), in which a nested structure of kernels is assumed. More precisely, given a family of kernel functions, we construct a series of nested kernel models with their associated parameters, or equivalently, a series of H nested functional bandwidths $\omega_{(1)}(t, \theta_{(1)}) \prec \cdots \prec \omega_{(H)}(t, \theta_{(H)})$. By $\omega_{(h)}(t, \theta_{(h)}) \prec \omega_{(h+1)}(t, \theta_{(h+1)})$ we denote that the bandwidth $\omega_{(h)}(t, \theta_{(h)})$ has parameters which are part of the parameters of the bandwidth $\omega_{(h+1)}(t, \theta_{(h+1)})$.

Example 3.2. Consider the family of piecewise constant functions with 3 pieces, in (7). We have that $\omega_{(1)}(t, \theta_{(1)}) = \omega_1$, with $\theta_{(1)} = \omega_1$, $\omega_{(2)}(t, \theta_{(2)}) = \omega_1 I_{[0,\tau_1]} + \omega_2 I_{(\tau_1,T]}$, with $\theta_{(2)} = (\omega_1, \omega_2, \tau_1)$, and finally $\omega_{(3)}(t, \theta_{(3)}) = \omega_1 I_{[0,\tau_1]} + \omega_2 I_{(\tau_1,\tau_2]} + \omega_3 I_{(\tau_2,T]}$, with $\theta_{(3)} = (\omega_1, \omega_2, \omega_3, \tau_1, \tau_2)$. Here $I_{[r,r']}$ denotes the indicator function, i.e., the function which is equal to 1 in the interval $[r, r']$ and 0 otherwise.

The idea of using nested models is to take advantage of the easy-to-tune structure of the elementary models and consider them as a simplification of the complex models.

When solving Problem (8) for $\omega_{(H)}(t, \theta_{(H)})$ we will use a sequential approach where the (suboptimal) solution obtained when using $\omega_{(h)}(t, \theta_{(h)})$, will be used as an initial solution of Problem (8) with $\omega_{(h+1)}(t, \theta_{(h+1)})$.

Example 3.3. For the bandwidth given in (7), once we have obtained the (suboptimal) solution of $\omega_{(h)}(t, \theta_{(h)})$ by $\theta^{opt}_{(h)} = (\omega^{opt}_1, \ldots, \omega^{opt}_h, \tau^{opt}_1, \ldots, \tau^{opt}_{h-1})$, we randomly select an interval $[\tau_{\ell-1}, \tau_\ell)$ and split it into two pieces by its midpoint, assigning the same bandwidth value to such two new pieces. In other words, the initial point of the parameters in the level $h+1$ turns out to be $\theta_{(h+1)} = (\omega^{opt}_1, \ldots, \omega^{opt}_{\ell-1}, \omega^{opt}_\ell, \omega^{opt}_\ell, \omega^{opt}_{\ell+1}, \ldots, \omega^{opt}_h, \tau^{opt}_1, \ldots, \tau^{opt}_{\ell-1}, \frac{\tau^{opt}_\ell + \tau^{opt}_{\ell-1}}{2}, \tau^{opt}_\ell, \ldots, \tau^{opt}_h)$.

The pseudocode of the nested algorithm defined in Section 3.1, is shown in Algorithm 2.

Algorithm 2 Nested heuristic for parameter tuning.

Input: H, nested functional bandwidths $\omega_{(1)}(t, \theta_{(1)}) \prec \cdots \prec \omega_{(H)}(t, \theta_{(H)})$.
• Randomly split the sample s into s_1, s_2, s_3 and s_4.
for C in the grid **do**
Initialization:
• $h := 1$.
• Randomly select an initial solution $\theta_{(h)} \in \Theta_{(h)}$.
• Set $\theta := \theta_{(h)}$
while $h \leq H$ **do**
1. Using samples s_1 and s_2, run the Alternating Procedure of Algorithm 1 for $\omega(t, \theta_{(h)})$, starting from θ and yielding $\theta^{opt}_{(h)} = (\omega^{opt}_1, \ldots, \omega^{opt}_h, \tau^{opt}_1, \ldots, \tau^{opt}_{h-1})$ as solution.
2. Randomly select $\ell \in \{1, 2, \ldots, h\}$.
3. Set $\theta := (\omega^{opt}_1, \ldots, \omega^{opt}_{\ell-1}, \omega^{opt}_\ell, \omega^{opt}_\ell, \omega^{opt}_{\ell+1}, \ldots, \omega^{opt}_h, \tau^{opt}_1, \ldots, \tau^{opt}_{\ell-1}, \frac{\tau^{opt}_\ell + \tau^{opt}_{\ell-1}}{2}, \tau^{opt}_\ell, \ldots, \tau^{opt}_{h-1})$ and $h := h + 1$.
4. Evaluate the accuracy in 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 α and $\theta_{(h)}$ are the ones associated to the optimal parameter C.
Output: optimal parameters C, $\theta^{opt}_{(h)}$, $\forall h$, the associated classification coefficients α, and the accuracy estimated from s_4.

3.3. Choice of the number of pieces, H

Thus far we have assumed that the number H of pieces is given as input in the problem, and hence the results are dependent on H. The larger H is, the better the accuracy (in the training sample) since more flexibility is added to the model. However, if a too large value of H is chosen, the number of parameters involved in the problem increases considerably, and this may deteriorate the accuracy in the test sample.

Therefore, it is sensible to define a strategy to determine the best H. In this respect, standard criteria, such as BIC, AIC or ICL, (Akaike, 1974; Biernacki, Celeux, & Govaert, 2000; Schwarz, 1978) can be applied in the SVM context, as done in Claeskens, Croux, and Kerckhoven (2008) for instance. They proposed two new information criteria which are inspired, but not equal to AIC and BIC, with the aim of giving consistent selection criteria without much additional computational costs. In contrast, in this paper, we propose to keep the parameter H with the largest accuracy on the validation sample s_3.

4. Numerical experiments

This section details the experiments performed (Section 4.1) and the main characteristics of the data bases here considered (Section 4.2). Finally, Section 4.3 is devoted to the computational results obtained.

4.1. Description of the experiments

In this section, a detailed description of the experiments carried out to test our methodology is made. To obtain stable estimates, k-fold cross-validation has been used to evaluate the performance of the algorithm on different data sets. The number k of folds varies depending on the size of the database. For small databases, k is equal to the number of observations, i.e., we performed leave-one-out, whilst for large databases we take $k = 10$. A database is con-

Table 3
Real data description summary.

	#Records	#Points measurements	#Records label −1	#Records label +1
ECG	200	96	67	133
growth	93	31	54	39
gun	200	96	100	100
MCO	89	360	44	45
phoneme	200	150	100	100
phoneme_large	1717	256	1022	695
rain	35	365	15	20
regions	35	365	20	15
synthetic_magnitude	150	100	75	75
tecator	215	100	77	138
wine	111	234	54	57
yoga	306	426	150	156

sidered small here if and only if it has less than 100 observations. See Table 3.

Algorithm 2 is run k times, one per fold. Each time, the division into four independent samples s_1, s_2, s_3, and s_4 is done as explained in Section 3.1. The number of runs of the multi-start local search optimization method is set to five. The algorithm is run until the maximum number of iterations reached to ten, or when the difference between the objective values in two consecutive iterations is less than 10^{-5}. The functional bandwidth $\omega(t, \theta)$ is the piecewise constant function in (7) with $H = 8$. The regularization parameter C varies in the set $\{2^{-10}, \ldots, 2^{10}\}$. The parameters $\theta_{(h)}$ are in the set $\Theta_{(h)} = \{(\omega_1, \ldots, \omega_h, \tau_1, \ldots, \tau_{h-1}) : \omega_\ell \geq 2^{-4}, \ell = 1, \ldots, h, 0 \leq \tau_1 \leq \ldots \leq \tau_{h-1} \leq T\}$, $\forall h = 1, \ldots, 8$.

For comparison purposes, apart from the standard SVM, i.e., our approach with $H = 1$, we have run three supervised classification methods for functional data, available at the fda.usc library of R (Febrero-Bande & Oviedo de la Fuente, 2012), namely *classif.depth*, *classif.kernel*, *classif.knn* with the default parameters. In order to obtain a fair comparison, the accuracy obtained is estimated on the very same testing sample s_4 used in our approach.

Our algorithm is coded in R and is carried out on a cluster with 2 terabyte of RAM memory at 6.2 TFlops, running CentOS Linux 7.3. The code is available upon request.

4.2. Description of the data sets

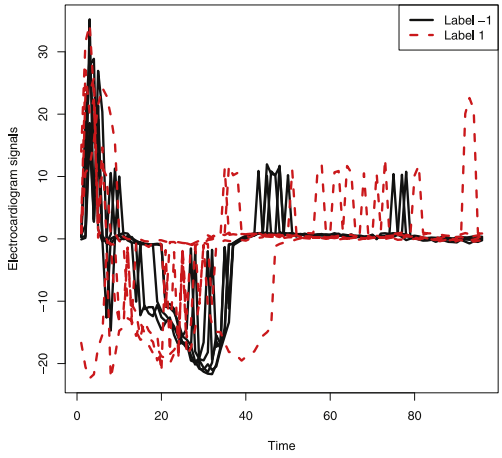
Our methodology has been tested in 12 benchmark data sets, widely used in the functional data classification literature, namely, *ECG*, (Chen et al., 2015; Xing, Pei, & Philip, 2009), *growth*, (Cuevas et al., 2007; Muñoz & González, 2010; Torrecilla Noguerales, 2015), *gun*, (Chen et al., 2015; Xing et al., 2009), *MCO*, (Baíllo et al., 2011; Cuevas, Febrero, & Fraiman, 2006; Ruiz-Meana et al., 2003) and Online companion of (Carrizosa et al., 2014), *phoneme*, (Ferraty & Vieu, 2006; Muñoz & González, 2010; Rossi & Villa, 2006; Torrecilla Noguerales, 2015), *phoneme_large*, (Berrendero et al., 2016; Delaigle & Hall, 2012; Friedman, Hastie, & Tibshirani, 2001a; 2001b), *rain*, (Martín-Barragán et al., 2014), *regions*, (Martín-Barragán et al., 2014), *synthetic_magnitude*, model 3 of (López-Pintado & Romo, 2009), *tecator*, (Ferraty & Vieu, 2006; Martín-Barragán et al., 2014; Rossi & Villa, 2006; Torrecilla Noguerales, 2015), *wine*, (Chen et al., 2015) and *yoga*, (Wei, 2006; Wei & Keogh, 2006). Note that the data set *phoneme* is used as described in the fda.usc library, (Febrero-Bande & Oviedo de la Fuente, 2012), of R. Table 3 summarizes the data sets description, which gives the overall number of records, the number of time measurements, and the number of records of each class. A plot with a sample of 10 instances of each data set is shown in Fig. 4, depicting in solid black line the observations with label −1 and in dashed red line the records with label 1. The number of folds is determined by leave-one-out in the data sets *growth*, *MCO*, *rain*, and *regions*, and with 10−fold cross-validation in the remaining databases.

4.3. Results

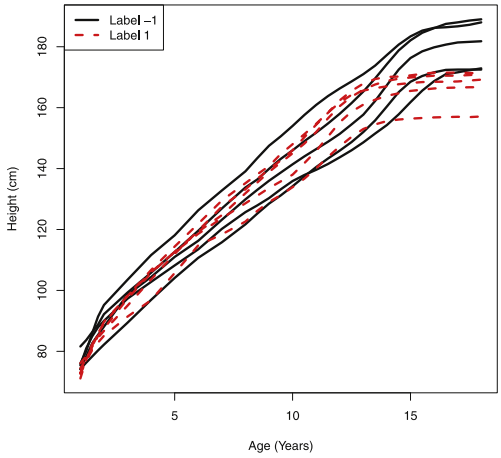
We provide the boxplots of the accuracy measured on s_4 from $h = 1$ to $h = 8$ for the different folds in the k-fold accuracy estimation procedure.

Boxplots are not very informative for small data sets, for which leave-one-out is performed. Indeed, for each fold either one obtains an accuracy of 0% or 100%, since either the testing observation is wrongly or correctly classified. For this reason, only the boxplots of the largest data sets are depicted. See Fig. 6. Moreover, the exact values of the average accuracy and its standard deviation, as well as the corresponding values for the three fda.usc library methods considered in Section 4.1, are also presented in Table 4 for the sake of comparison. The four gray columns correspond to the four methods we are comparing with, denoted as *depth*, *kernel*, *knn* and *classic SVM*, $h = 1$. Finally, last column of Table 4 gives the best number of pieces chosen, according to the strategy explained in Section 3.3.

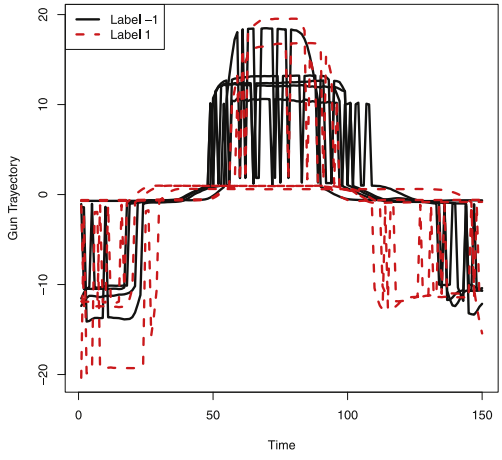
We have highlighted in bold in Table 4 the maximum of the accuracy values for $h = 2, \ldots, 8$ which are equal or greater than any of the four methods. In general, our method for $h = 2, \ldots, 8$ is better than the four comparative approaches in the data sets *growth*, *MCO*, *phoneme*, *phoneme_large*, and *regions*. This improvement may be produced by the shape of the curves. The different class labels seem to be easy to identify depending on the time subinterval, and therefore our strategy makes easier such separation. Observe for instance, the *growth* data set, in which the two classes have a different pattern around the time instant 15. Moreover, it is seen that the improvement in the accuracy strongly depends on the data set considered. Indeed, no improvement is seen for the databases *gun*, *rain*, and *tecator* when comparing our methodology with $h = 1$ and $h \geq 2$. However, for some of the values $h \geq 2$ the accuracy obtained in *gun* is better than that provided by *depth*. The results of our approach in the database *rain* are always better than the ones provided by the three fda.usc methods. In contrast, such three methods should be applied if the *tecator* data set is studied. In the databases *ECG*, *growth*, *phoneme_large* and *yoga* there is a minor improvement (about a 0.5%) when comparing the classic SVM with our approach for $h \geq 2$. Such improvement also holds in the *ECG* data set when comparing with the *depth* method. The accuracy value obtained in *phoneme_large* with our approach when $h = 4$ pieces are optimally chosen is better than all the three fda.usc methods. Analogous conclusions are obtained in the *yoga* data set. A considerably larger accuracy is obtained in databases *MCO*, *phoneme*, *regions*, *synthetic_magnitude*, and *wine* when solving the problem with $h \geq 2$ than when solving with $h = 1$, i.e., the classic SVM. In some cases such improvement yields



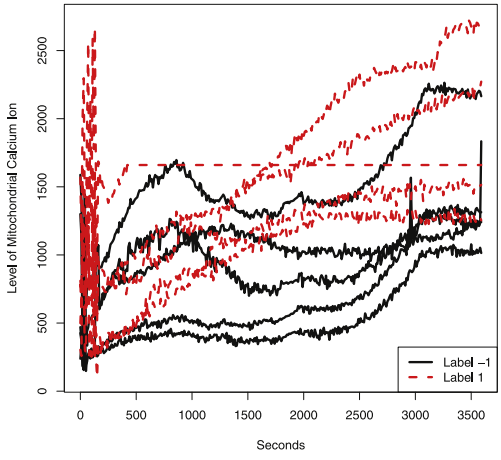
(a) ECG



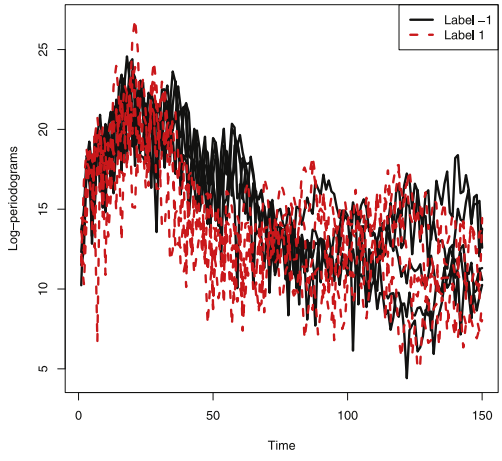
(b) growth



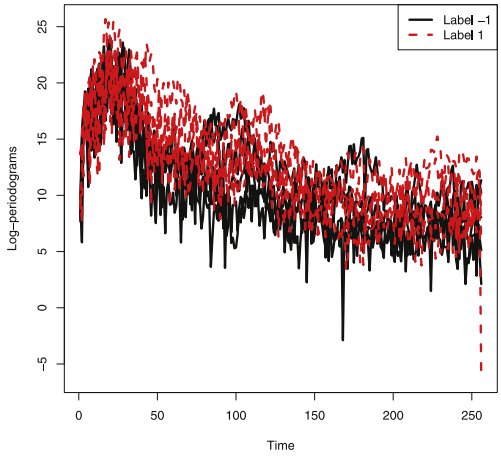
(c) gun



(d) MCO

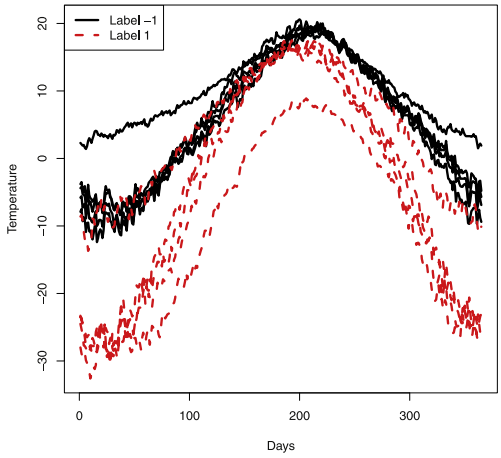


(e) phoneme

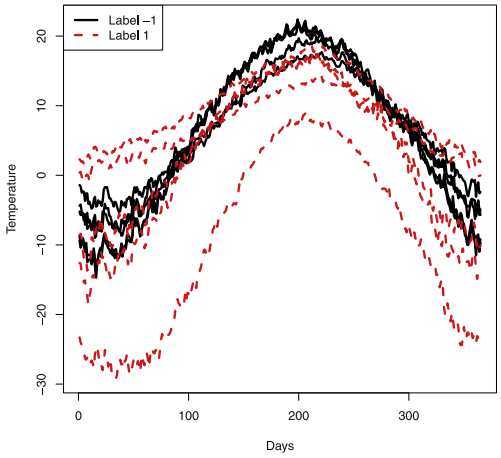


(f) phoneme_large

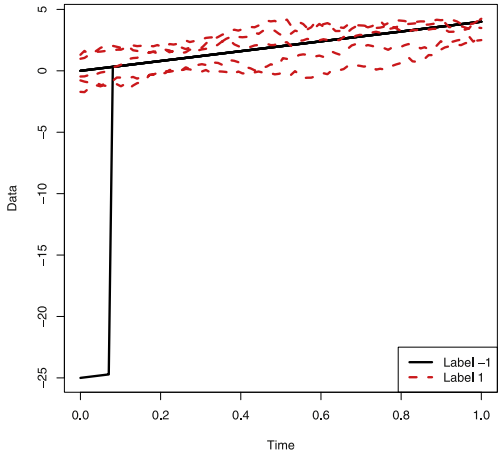
Fig. 4. Sample of functional data in the real data sets analyzed.



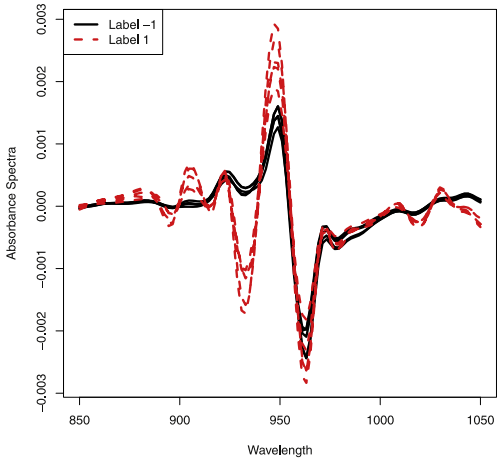
(g) rain



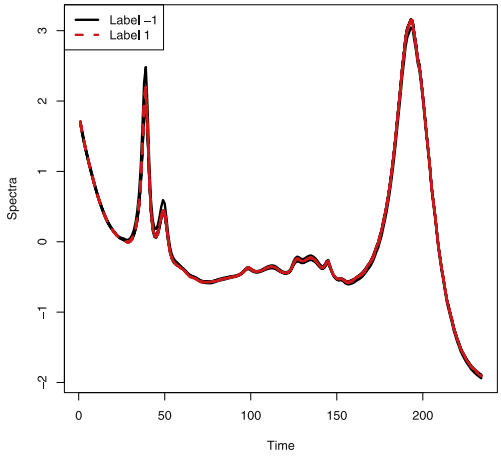
(h) regions



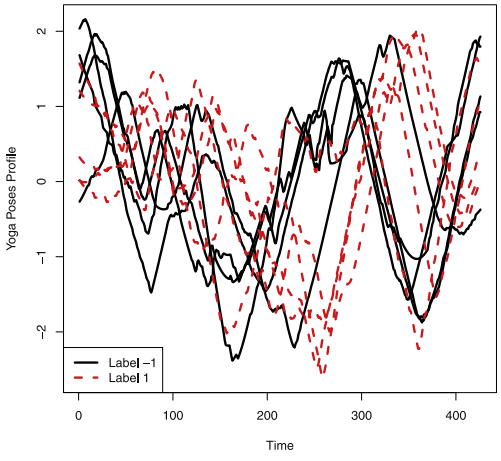
(i) synthetic_magnitude



(j) tecator



(k) wine



(l) yoga

Fig. 4. Continued

Table 4
Average and standard deviation (between brackets) of the accuracy estimated on sample s_4 for all the data sets after running the three methods available at fda.usc (depth, kernel and knn) and running our approach from $h = 1$ to $h = 8$. It is highlighted in bold the maximum of the accuracy values for $h = 2, \ldots, 8$ which are equal or greater than any of the four methods. Last column presents the average value of the best parameter H.

	Depth	Kernel	knn	Classic SVM, $h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$	$h = 7$	$h = 8$	Best H
ECG	67.51 (8.00)	74.41 (8.02)	83.92 (11.51)	67.51 (4.60)	67.51 (4.60)	67.51 (4.60)	**68.01 (6.12)**	**68.01 (6.12)**	**68.01 (6.12)**	**68.01 (6.12)**	**68.01 (6.12)**	1.7
growth	78.49 (41.30)	94.62 (22.67)	92.47 (26.52)	94.62 (22.67)	94.62 (22.67)	94.62 (22.67)	94.62 (22.67)	93.54 (24.70)	**95.69 (20.39)**	94.62 (22.67)	94.62 (22.67)	1.3
gun	52.50 (5.40)	73.50 (8.51)	78.00 (13.37)	69.50 (14.61)	68.00 (14.75)	**69.50 (13.83)**	**69.50 (13.83)**	67.50 (16.20)	67.00 (12.73)	69.00 (14.29)	**69.50 (13.63)**	3
MCO	67.41 (47.13)	78.65 (41.20)	79.77 (40.39)	80.89 (39.53)	78.65 (41.20)	83.14 (37.64)	82.02 (38.61)	83.14 (37.64)	**86.51 (34.34)**	83.14 (37.64)	83.14 (37.64)	4.15
phoneme	76.00 (6.58)	80.00 (8.16)	73.50 (11.55)	80.50 (9.55)	81.50 (9.44)	81.00 (9.66)	81.50 (9.44)	**82.00 (9.77)**	81.00 (8.75)	80.50 (9.55)	80.50 (8.95)	4
phoneme_large	71.69 (3.21)	65.46 (1.52)	78.91 (2.84)	81.24 (2.39)	81.47 (1.98)	81.24 (1.68)	**82.00 (2.11)**	81.65 (2.40)	81.59 (2.40)	81.83 (2.39)	81.65 (2.32)	2.1
rain	82.85 (38.23)	80.00 (40.58)	77.14 (42.60)	82.85 (38.23)	80.00 (40.58)	80.00 (40.58)	**82.85 (38.23)**	80.00 (40.58)	80.00 (40.58)	77.14 (42.60)	77.14 (42.60)	1.91
regions	77.14 (42.60)	85.71 (35.50)	77.14 (42.60)	82.84 (38.23)	88.56 (32.28)	88.57 (32.28)	88.57 (32.28)	85.71 (35.50)	**91.42 (28.40)**	**91.42 (28.40)**	88.57 (32.28)	1.77
synthetic_magnitude	99.37 (1.97)	55.97 (5.20)	96.54 (3.53)	90.40 (13.55)	91.11 (11.66)	90.40 (13.55)	**92.54 (10.80)**	**92.54 (10.80)**	**92.54 (10.80)**	**92.54 (10.80)**	**92.54 (10.80)**	1.9
tecator	94.87 (4.70)	98.13 (2.40)	98.11 (3.30)	74.04 (15.88)	**74.04 (15.88)**	**74.04 (15.88)**	**74.04 (15.88)**	**74.04 (15.88)**	**74.04 (15.88)**	73.59 (15.25)	**74.04 (15.88)**	1.8
wine	61.87 (17.50)	91.93 (6.26)	92.03 (7.57)	71.27 (19.06)	75.77 (14.76)	**77.59 (16.59)**	77.59 (16.98)	75.68 (17.95)	76.68 (16.59)	76.68 (16.59)	76.68 (16.59)	2.6
yoga	84.13 (7.01)	96.08 (4.01)	96.73 (3.46)	95.11 (3.13)	95.45 (3.47)	95.76 (3.07)	**96.44 (3.20)**	95.78 (3.41)	96.09 (3.67)	96.09 (3.67)	96.09 (3.67)	3.6

around ten percentage points of difference in accuracy terms. Such a large accuracy also occurs when comparing our approach with the three fda.usc methods in the databases *MCO*, *regions* and *wine*. The improvement is not so evident in the *phoneme* data set. In the data set *synthetic_magnitude*, our results are comparable to provided by *depth* and *knn*, but much better than the ones in *kernel*.

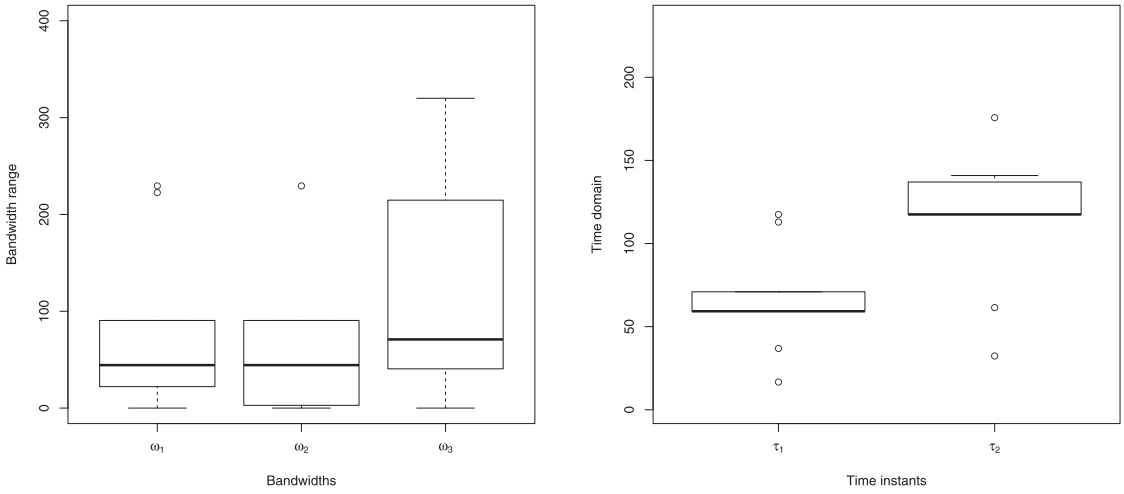
Apart from the improvements in the accuracy, our approach enables us to identify subintervals of special interest. This fact would be impossible if the standard scalar bandwidth, which treats equally all time instants, were considered. We highlight, for instance, the case of the *wine* data set, whose curves are almost identical except around the time instants at which peaks occur. Fig. 5 shows the boxplots of the values of ω_1, ω_2, ω_3, τ_1 and τ_2 obtained when a functional bandwidth with $h = 3$ pieces is sought. We observe that the time instants which distinguish one piece from another are around 50 and 125, which coincides with the points of some of the peaks. Furthermore, the associated weight is greater in the third part, where the biggest peak is located.

Regarding the trajectory of the accuracy versus the number of pieces, we observe that there is not a clear pattern in the behavior. For instance, in the *MCO* data set, we have worse results with $h = 2$ pieces than with the classic SVM ($h = 1$). However, a difference of six points is obtained when comparing $h = 6$ with $h = 1$.

In contrast, in the *regions* data set, the accuracy with $h \geq 2$ are significantly better than with $h = 1$, reaching the maximum value with $h = 6$. Similar conclusions can be drawn in the remaining data sets.

This fact shows the importance of using an adequate value of H. Since the value of the parameter H depends on the division of the data set, we show in the last column of Table 4 the average value of the best H parameter estimated on sample s_3 as explained in Section 3.3.

With respect to the running times, we first point out that most of the time is spent in the training phase since once the classifier is built, classifying new observations is definitely quick. Indeed, it just reduces to compute the score given in (1) and follow the corresponding classification rule. Moreover, for a given fold, for a fixed value of C, h and for each iteration of the alternating approach, the resolution of both optimization problems (2) and (9) highly depends on the size of the data set. Particularly, solving one SVM problem in the *rain* data set lasts an average of 0.3 seconds, whereas 3.3 seconds are spent if Problem (2) is solved on the *phoneme_large* data set. On the other hand, the average running time of Problem (9) goes from 0.5 seconds to 26.8. Such values correspond respectively to the *rain* and *phoneme_large* databases. Note also that the computational times will depend on the value of h since harder optimization problems of type (9), involving more decision variables, are to be solved as h grows. For example, in the data set *yoga*, 5.9 seconds are spent in solving our approach with the single bandwidth case, i.e., $h = 1$ and 7.5 when $h = 2$ pieces are sought. In order to have the whole amount of time invested in our algorithm, we should take into account different elements, such as the number of folds, the number of C values in the grid, the maximum number of iterations in the alternating approach, and the number of runs in the multi-start. Nevertheless, the total time does not increase linearly, since running the code in parallel, as done in this paper, reduces the elapsed time. Furthermore, our strategy of nesting the problem alleviates the increase in running times since the optimization of the most complex models is not started from scratch but from the optimal solution of the simplest models. The running times of the three methods of the fda.usc library with the default parameters are around 3 seconds, for a given fold. However, our approach gains in interpretability terms, as has already been mentioned.



(a) Bandwidth (b) Time instants

Fig. 5. Bandwidth and time instants results for the *wine* data set.

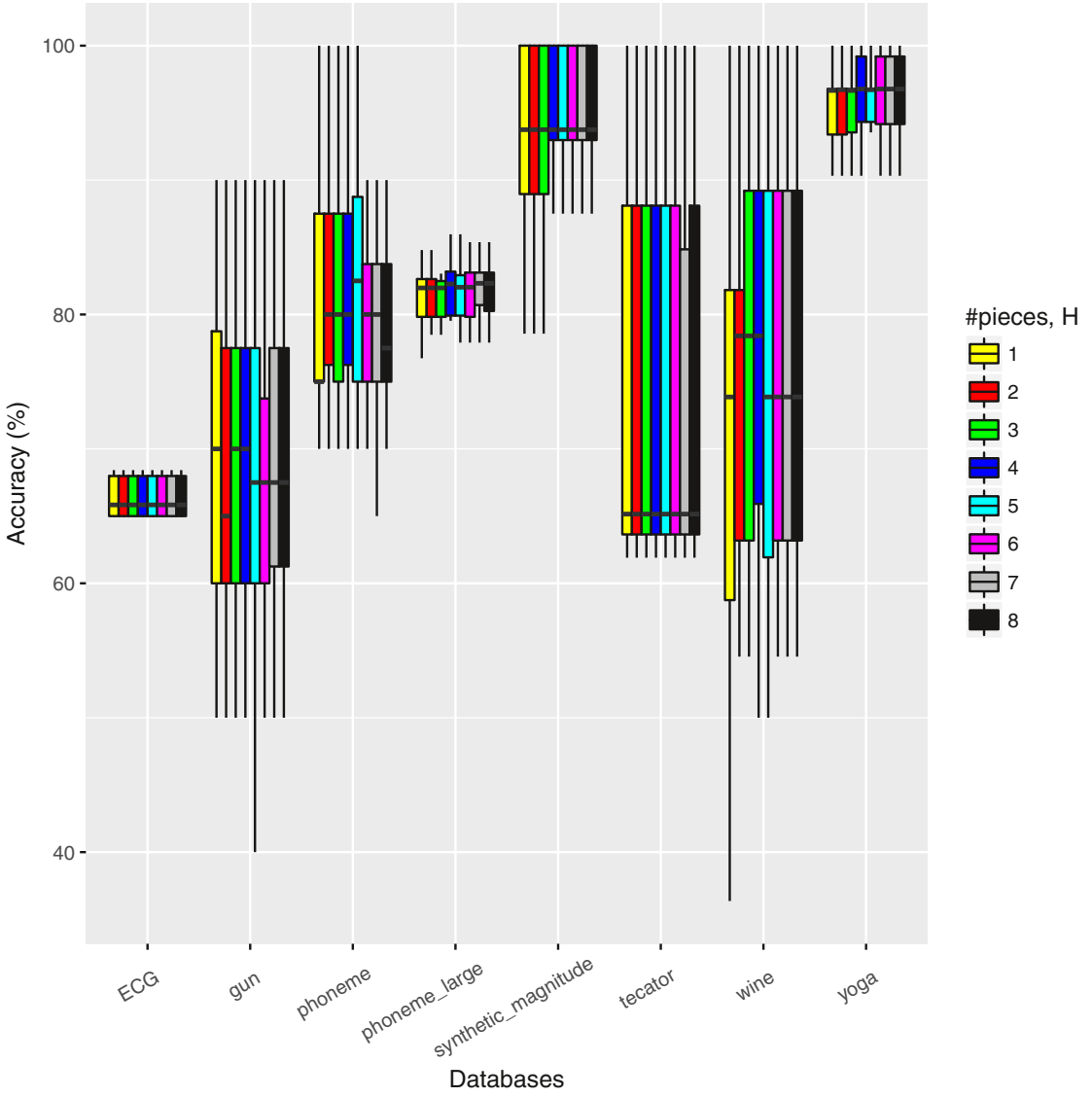


Fig. 6. Accuracy boxplots in the analyzed larger data sets depending on the number of pieces, *H*. Since the boxplots are rather informative for the small data sets, i.e., *growth*, *MCO*, *rain* and *regions*, only the accuracy values of the remaining databases are depicted.

5. Conclusions and extensions

In this paper, we have shown how SVM for functional data can be easily improved if a functional bandwidth, to be tuned via a nested heuristic, is used. By using very simple functional parameters, together with our tuning procedure, we obtained better accuracy than with the traditional scalar parameter model in the test sets. The methodology here proposed is able to identify the critical points in which a change in the behavior of the functions is produced, yielding the most relevant intervals in terms of the classification rate and also regarding the interpretability of the results.

The difficulties associated to the tuning of more complex structures are mitigated by the use of a heuristic that exploits the nested structure of the functional parameter, by using the (suboptimal) solution of one level as an initial solution for the next level. Our tuning procedure takes advantage of the functional nature of the data by expressing the tuning problem as a bilevel optimization problem in continuous variables. In contrast to the usual approach, where the misclassification rate is minimized, here the correlation between labels and scores are optimized, allowing us to use gradient-based local search algorithms.

In our approach, the number of pieces of the functional bandwidth, H, is fixed from the beginning, and the trajectory of the classification rates for the different number of pieces is shown. However, since the results depend on H we choose the value of H yielding the best accuracy, as estimated on the validation sample. The analysis performed here, using piecewise constant functions as bandwidths, can be easily extended to other expressions such as polynomials, or piecewise polynomials, including splines (De Boor, 1978; Friedman et al., 2001b). Apart from the Pearson coefficient, different types of association measures can be applied (Székely et al., 2007; Torrecilla Noguerales, 2015).

The functional data here considered are univariate functions. The case of multivariate (hybrid) functional data, (Jiménez-Cordero & Maldonado, 2018) can also be addressed with our proposal, after the convenient modification of the kernel function.

The standard hinge loss function has been used in the SVM. Our approach might also be adapted to other loss functions, such as the so-called ramp loss, (Brooks, 2011), by replacing (2) with the corresponding SVM problem. The same happens if the SVM in (2) is replaced by some related methods such as the least-squares SVM, e.g., (Cruz-Cano, Chew, Choi, & Leung, 2010).

We have limited ourselves to classification problems. If instead, functional regression is pursued, (Sood, James, & Tellis, 2009), our methodology can be easily adapted to this context, replacing SVM by Support Vector Regression. This research line is also under investigation.

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. The team thanks the Scientific Computing Center of Andalucía (CICA) for the computing services provided.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.

Baíllo, A., Cuevas, A., & Cuesta-Albertos, J. A. (2011). Supervised classification for a family of gaussian functional models. *Scandinavian Journal of Statistics, 38*(3), 480–498.

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

Biau, G., Bunea, F., & Wegkamp, M. H. (2005). Functional classification in Hilbert spaces. *IEEE Transactions on Information Theory, 51*(6), 2163–2172.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(7), 719–725.

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., & Martín-Barragán, B. (2017). Variable Selection in Classification for Multivariate Functional Data. *Technical Report*. University of Edinburgh – Universidad de Sevilla. Available at https://www.researchgate.net/publication/321400055_Variable_Selection_in_Classification_for_Multivariate_Functional_Data.

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.

Brooks, J. P. (2011). Support vector machines with the ramp loss and the hard margin loss. *Operations Research, 59*(2), 467–479.

Bugeau, A., & Pérez, P. (2007). Bandwidth selection for kernel estimation in mixed multi-dimensional spaces.

Cai, Z., Fan, J., & Yao, Q. (2000). Functional-coefficient regression models for nonlinear time series. *Journal of the American Statistical Association, 95*(451), 941–956.

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.

Carrizosa, E., & Romero Morales, D. (2013). Supervised classification and mathematical optimization. *Computers & Operations Research, 40*(1), 150–165.

Cauwenberghs, G., & Poggio, T. (2001). Incremental and decremental support vector machine learning. In *Advances in neural information processing systems* (pp. 409–415).

Chen, Q., Wynne, R., Goulding, P., & Sandoz, D. (2000). The application of principal component analysis and kernel density estimation to enhance process monitoring. *Control Engineering Practice, 8*(5), 531–543.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (2015). The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

Claeskens, G., Croux, C., & Kerckhoven, J. V. (2008). An information criterion for variable selection in support vector machines. *Journal of Machine Learning Research, 9*(Mar), 541–558.

Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research, 153*(1), 235–256.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Cruz-Cano, R., Chew, D. S., Choi, K.-P., & Leung, M.-Y. (2010). Least-squares support vector machine approach to viral replication origin prediction. *INFORMS Journal on Computing, 22*(3), 457–470.

Cuevas, A., Febrero, M., & Fraiman, R. (2006). On the use of the bootstrap for estimating functions with functional data. *Computational Statistics & Data Analysis, 51*(2), 1063–1074.

Cuevas, A., Febrero, M., & Fraiman, R. (2007). Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics, 22*(3), 481–496.

De Boor, C. (1978). A practical guide to splines. *Applied Mathematical Sciences*: 27. Springer-Verlag New York.

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

Duong, T., Cowling, A., Koch, I., & Wand, M. (2008). Feature significance for multivariate kernel density estimation. *Computational Statistics & Data Analysis, 52*(9), 4225–4242.

Febrero-Bande, M., & Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: the r package fda.usc. *Journal of Statistical Software, 51*(4), 1–28.

Ferraty, F., & Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Ferris, M. C., & Munson, T. S. (2004). Semismooth support vector machines. *Mathematical Programming, 101*(1), 185–204.

Friedman, J., Hastie, T., & Tibshirani, R. (2001a). Datasets for *The Elements of Statistical Learning*. https://web.stanford.edu/~hastie/ElemStatLearn/data.html.

Friedman, J., Hastie, T., & Tibshirani, R. (2001b). The elements of statistical learning. *Springer Series in Statistics*: 1. Springer, Berlin.

Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks, 15*(8), 1059–1068.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics, 36*(3), 1171–1220.

Jiménez-Cordero, A., & Maldonado, S. (2018). Automatic feature scaling and selection for support vector machine classification with functional data. *Technical Report*. Universidad de los Andes - Universidad de Sevilla. Available at https://www.researchgate.net/publication/323428879_Automatic_Feature_Scaling_and_Selection_for_Support_Vector_Machine_Classification_with_Functional_Data.

Kadri, H., Duflos, E., Preux, P., Canu, S., & Davy, M. (2010). Nonlinear functional regression: a functional RKHS approach. In *Proceedings of the International conference on artificial intelligence and statistics* (pp. 374–380).

Kästner, M., Hammer, B., Biehl, M., & Villmann, T. (2012). Functional relevance learning in generalized learning vector quantization. *Neurocomputing, 90*, 85–95. Advances in artificial neural networks, machine learning, and computational intelligence (ESANN 2011).

Keerthi, S. S., & Lin, C.-J. (2003). Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation, 15*(7), 1667–1689.

Laukaitis, A., & Račkauskas, A. (2005). Functional data analysis for clients segmentation tasks. *European Journal of Operational Research, 163*(1), 210–216.

Lessmann, S., & Voß, S. (2009). A reference model for customer-centric data mining with support vector machines. *European Journal of Operational Research, 199*(2), 520–530.

López-Pintado, S., & Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association, 104*(486), 718–734.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research, 261*(2), 656–665.

Maldonado, S., Weber, R., & Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences, 181*(1), 115–128.

Martín-Barragán, B., Lillo, R., & Romo, J. (2014). Interpretable support vector machines for functional data. *European Journal of Operational Research, 232*(1), 146–155.

Muñoz, A., & González, J. (2010). Representing functional data using support vector machines. *Pattern Recognition Letters, 31*(6), 511–516.

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

Ramsay, J. O., & Silverman, B. W. (2002). Applied functional data analysis: methods and case studies. *Springer Series in Statistics*: 77. Springer-Verlag.

Ramsay, J. O., & Silverman, B. W. (2005). Functional data analysis. *Springer Series in Statistics* (2nd). Springer-Verlag.

Richtárik, P., & Takáč, M. (2016). Parallel coordinate descent methods for big data optimization. *Mathematical Programming, 156*(1–2), 433–484.

Rossi, F., & Villa, N. (2006). Support vector machine for functional data classification. *Neurocomputing, 69*(7), 730–742.

Rossi, F., & Villa, N. (2008). Recent advances in the use of SVM for functional data classification. In *Functional and operatorial statistics* (pp. 273–280). Heidelberg: Physica-Verlag HD.

Ruiz-Meana, M., Garcia-Dorado, D., Pina, P., Inserte, J., Agulló, L., & Soler–Soler, J. (2003). Cariporide preserves mitochondrial proton gradient and delays atp depletion in cardiomyocytes during ischemic conditions. *American Journal of Physiology-Heart and Circulatory Physiology, 285*(3), H999–H1006.

Sain, S. R. (2002). Multivariate locally adaptive density estimation. *Computational Statistics & Data Analysis, 39*(2), 165–186.

Sato, A., & Yamada, K. (1996). Generalized learning vector quantization. In *Advances in neural information processing systems* (pp. 423–429).

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*(2), 461–464.

Sood, A., James, G. M., & Tellis, G. J. (2009). Functional regression: A new model for predicting market penetration of new products. *Marketing Science, 28*(1), 36–51.

Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*(3), 293–300.

Székely, G. J., Rizzo, M. L., Bakirov, N. K., et al. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics, 35*(6), 2769–2794.

Torrecilla Noguerales, J. L. (2015). *On the theory and practice of variable selection for functional data*. Universidad Autónoma de Madrid (Ph.D. thesis).

Tuddenham, R. D., & Snyder, M. M. (1954). *Physical growth of california boys and girls from birth to eighteen years.* (1, pp. 183–364). Publications in Child Development. University of California, Berkeley.

Wang, H., & Yao, M. (2015). Fault detection of batch processes based on multivariate functional kernel principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 149*, 78–89.

Wang, H., Zheng, B., Yoon, S. W., & Ko, H. S. (2018). A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research, 267*(2), 687–699.

Wang, J.-L., Chiou, J.-M., & Müller, H.-G. (2016). Functional Data Analysis. *Annual Review of Statistics and Its Application, 3*(1), 257–295.

Wei, L. (2006). http://alumni.cs.ucr.edu/~wli/selfTraining/.

Wei, L., & Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the twelfth ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD '06* (pp. 748–753). ACM.

Wu, C. O., Chiang, C.-T., & Hoover, D. R. (1998). Asymptotic confidence regions for kernel smoothing of a varying-coefficient model with longitudinal data. *Journal of the American Statistical Association, 93*(444), 1388–1402.

Xing, Z., Pei, J., & Philip, S. Y. (2009). Early prediction on time series: A nearest neighbor approach. *IJCAI*, 1297–1302.