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

DOI: 10.1016/j.ejor.2012.08.017

Computational Intelligence & Information Management

Interpretable support vector machines for functional data

Belen Martin-Barragan *, Rosa Lillo, Juan Romo

Department of Statistics, Universidad Carlos III de Madrid, Spain

ARTICLE INFO

Article history:
Received 8 November 2011
Accepted 18 August 2012

Keywords:
Data mining
Interpretability
Classification
Linear programming
Regularization methods
Functional data analysis

ABSTRACT

Support Vector Machines (SVMs) is known to be a powerful nonparametric classification technique even for high-dimensional data. Although predictive ability is important, obtaining an easy-to-interpret classifier is also crucial in many applications. Linear SVM provides a classifier based on a linear score. In the case of functional data, the coefficient function that defines such linear score usually has many irregular oscillations, making it difficult to interpret.

This paper presents a new method, called *Interpretable Support Vector Machines for Functional Data*, that provides an interpretable classifier with high predictive power. Interpretability might be understood in different ways. The proposed method is flexible enough to cope with different notions of interpretability chosen by the user, thus the obtained coefficient function can be sparse, linear-wise, smooth, etc. The usefulness of the proposed method is shown in real applications getting interpretable classifiers with comparable, sometimes better, predictive ability versus classical SVM.

1. Introduction

The term Functional Data Analysis was already used in [30] two decades ago. Since them, especially in the last decade, it has become a fruitful field in statistic. The range of real world applications where the objects can be thought as functions is as diverse as speech recognition, spectrometry, meteorology or clients segmentation, to cite just a few [19,9,17,20]. The objects of study in Functional Data Analysis (FDA) are functions. A good review of different FDA techniques applied to real world problems can be found in [31]. For a deeper insight into the subject see, e.g., [10,32].

We deal with the problem of classifying functional data. Suppose we observe a binary response Y (the class) to a functional predictor X, where $X \in \mathcal{X}$ is a function defined on the bounded interval $\mathcal{I}$, i.e., $X : \mathcal{I} \mapsto \mathbb{R}$, and $\mathcal{X}$ is a given set of functions. Our aim is to construct a classification rule that predicts Y for a given functional datum X with good prediction ability and some interpretability properties.

The classification rule is based on the sign of the so-called *score function* f. The score function is an operator $f : \mathcal{X} \mapsto \mathbb{R}$ that assigns a real number to a given function X. Since our aim is interpretability, we consider the score function to be a linear operator $T_{\beta,\omega}$ with coefficient function $w \in \mathcal{X}$ and intercept $\beta \in \mathbb{R}$,

$$f(X) = T_{\beta,w}X = \int_{\mathcal{I}} w(t)X(t)dt + \beta = \langle w, X\rangle + \beta, \tag{1}$$

where $\langle f, g\rangle = \int_{\mathcal{I}} f(t)g(t)dt$. The estimation of the coefficient function w on the whole interval $\mathcal{I}$ is an infinite dimensional problem. This issue is addressed via regularization, which simultaneously allows us to address our other concern: interpretability.

As in standard Support Vector Machines (SVMs), $w(t)$ express the discriminative power of $X(t)$. For example, areas where $w(t)$ is zero or small has none or low discrimination power, whereas for $|w(t)|$ large, one can expect the behavior of $X(t)$ to have influence over the classification. This idea provides a clear interpretation of $w(t)$ at a particular time point t, but getting a general idea about the coefficient function w requires it to be simple: cases where $w(t)$ has unnatural wiggles all along the interval $\mathcal{I}$ are difficult to interpret.

The simplicity of w might be understood in different ways depending on the application. For instance, a coefficient function that is non-zero in just a few points, could detect the few points that are more relevant in classification. This idea has been proposed within a logistic regression model, see [24]. In other situations, one might prefer a coefficient function that is constant over a few subintervals of $\mathcal{I}$ and zero on the rest. A method that detects a few segments with high discriminative power has been proposed in [22] by combining feature selection, classical linear discriminant analysis and SVM. In gene expression analysis, detection of relevant segments are also quite desirable because relevant genes are expected to be located close to each other along the chromosome [33]. All this literature provides different methodologies for different notions of interpretability. Our proposal is to provide a common framework where all this notions can be seen as particular cases.

We use the interpretability notions proposed by [17] for functional linear regression. We consider that a classifier is interpretable

* Corresponding author.
E-mail addresses: belen.martin@uc3m.es (B. Martin-Barragan), lillo@est-econ.uc3m.es (R. Lillo), juan.romo@uc3m.es (J. Romo).

if one or several derivatives of the coefficient function w are sparse, i.e., the derivatives are zero in many points. The choice of the derivatives that are enforced to be sparse depends on the notion of interpretability preferred by the practitioner. In this context, this paper proposes a new method, that we call *Interpretable Support Vector Machines for Functional Data* (ISVMFD), producing SVM-based classifiers for functional data which have high classification accuracy and whose coefficient functions are easy to interpret. The problem is formulated as a linear program, in the framework of L_1-norm SVM.

The seek of interpretability is not new in functional data analysis. A penalized version of the classical Linear Discriminant Analysis (LDA) is proposed in [14] and is denoted as PDA. PDA and ISVMFD share common ideas: regularization and interpretability. However the two methods are different in many aspects. The main difference is the error criteria used: ISVMFD is based in minimizing the hinge loss whereas PDA is based on maximizing the between-class variance relative to the within-class variance. Besides, interpretability in PDA is achieved by using a penalty matrix that imposes a spatial smoothness constraint on the coefficients.

Another approach for finding interpretable classifier is variable clustering techniques, or in a more general framework, variable selection. Methods that use this kind of selection techniques are usually based on a two-phase framework. There is a phase where the variables are clustered or selected, and the classifier is built in a posterior phase. For instance, in [18] a variable clustering phase is embedding into a three-phase classification procedure in order to select ranges in spectra. See, for instance, [12,13] for a review in the wide variety of feature selection methods that can be applied within a two-phase framework. In contrast, when IFSVM is used, the selection phase is done together with the construction of the classifier.

The outline of the paper is as follows: Section 2 reviews classical literature for SVM on multivariate data, its extension to functional data and how interpretability has been addressed for multivariate data. In Section 3 the ISVMFD method is introduced and a proposal to implement it through the use of a basis is provided. Section 4 studies how other methods available in the literature are particular cases of ISVMFD. A wide study with two real-world datasets is presented in Section 5 and finally, in Section 6, several conclusions are driven. An Online Companion Appendix that includes more illustrative examples is provided.

2. Support vector machines

We focus in this paper on the binary supervised classification problem, where two classes $\{-1,1\}$ of curves need to be discriminated. SVM [8,27,38] have become very popular during the last decade. The basic idea behind SVM can be explained geometrically. If the data are living in a p-dimensional space, SVM finds the separating hyperplane with maximal margin, i.e., the one furthest away from the closest objects. This geometrical problem is expressed as a smooth convex problem with linear constraints, solved either in its primal or dual form. Another interpretation can be done in terms of the regularization theory where the hinge loss plus a quadratic regularization penalty is minimized [15,35]. The most popular and powerful versions of SVM embed the original variables into a higher dimensional space [16]. This embedding is usually implicitly specified by the choice of a function called kernel.

Extensions of SVM to functional data have been proposed in [28,34]. In [28], SVM is used to represent the functional data by projecting the original functions onto the eigenfunctions of a Mercer Kernel. Ref. [34] define new classes of kernels that take into account the functional nature of the data. Two types of functional kernels are proposed: projection-based kernels and transformation-based kernels. In projection-based kernels, the idea is to reduce the dimensionality of the input space, i.e., to apply the standard filtering approach of FDA. Transformation-based kernels allow to take into account expert knowledge (such as the fact that the curvatures of a function can be more discriminant than its values).

In the multivariate context, kernels provide an implicit way to get a nonlinear classifier, by projecting the data on the higher dimensional space induced by the kernel. The final classifier is nonlinear in the original space, but linear in the projected space. Functional data are indeed high dimensional and the high dimensionality usually generates problems. Therefore the use of kernels to project data on a higher dimensional space seems to be less crucial. Moreover, the kernel-based classifier would be easy to interpret in the projected space, but not in the original one. We focus on the linear kernel in our method.

The interpretability issue in SVM has already been addressed for multivariate data. The first attempts to make SVM more interpretable make use of a two-step procedure: first, SVM is run, and then a rule, resembling the SVM-classifier but easier to interpret, is built. See, e.g. [1,3,26,25]. One obtains an alternative classifier which hopefully get similar predictions, but is more interpretable. Recently, a two-stage iterated method is proposed for credit decision making [23], which combines feature selection and multi-criteria programming. In [6,7], one-step SVM-based procedures are proposed to get the relevant variables and the relevant interactions between variables. Although one would expect classification rates to deteriorate when looking for interpretable classifiers, the experiments in [6,7] show that their proposals are competitive with SVM. See [2,21,37,39] for other recent references on the topic.

3. Methodology

3.1. Interpretable support vector machines for functional data

Let $\{X_u, Y_u\}_{u=1}^n$ be a sample of n functional data $X_u \in \mathcal{X}$ together with its class $Y_u \in \{-1,1\}$. The classical SVM with the linear kernel seeks for the coefficient function w that minimizes

$$min_{w,\beta}\|w\|_q^q + C\sum_{u=1}^{n} h(y_u, \langle w, X_u\rangle + \beta) \tag{2}$$

where $\|\cdot\|_q$ is the q-norm, $h(y,s) = (1-ys)_+$ is the hinge loss and C is a tuning parameter that trades off the regularization term $\|\omega\|_q^q$ and the loss term.

The class is predicted as the sign of the score function given in (1). In case of ties, i.e., $f(X) = 0$, prediction can be randomly assigned or following some predefined order. Throughout this article, following a worst case approach, ties will be considered as misclassifications.

Although the regularization with the Euclidean norm is the most common, other norms have also been applied. For instance, the L_1 norm is known to be good when a sparse coefficient vector is desirable. Ref. [4] demonstrated the usefulness of penalties based on the L_1 norm in classification problems. In regression, LASSO [35] and the Dantzig selector [5] also successfully use the L_1 norm in high-dimensional problems.

In order to get the interpretable classifier, we propose a modified version of SVM that we call Interpretable Support Vector Machines for Functional Data (ISVMFD). Following the concepts of interpretability described in Section 1, we propose to use a different regularization term that depends on the preferences of the user for the interpretability notion. The user must select one or several derivatives to be sparse. For example, if the user is concerned with detecting relevant time points, the zero derivative (the actual w) is selected to be sparse. Sparsity of the first derivative leads to constant-wise w which is useful to identify relevant segments. A user might prefer a coefficient function that is zero over large regions,

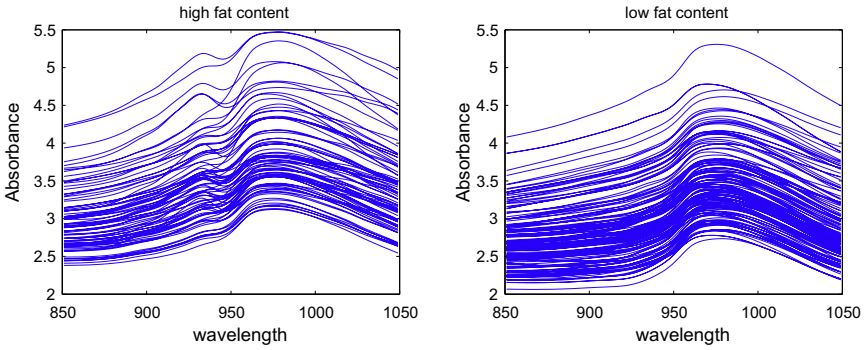

Fig. 1. Original data for `tecator` dataset.

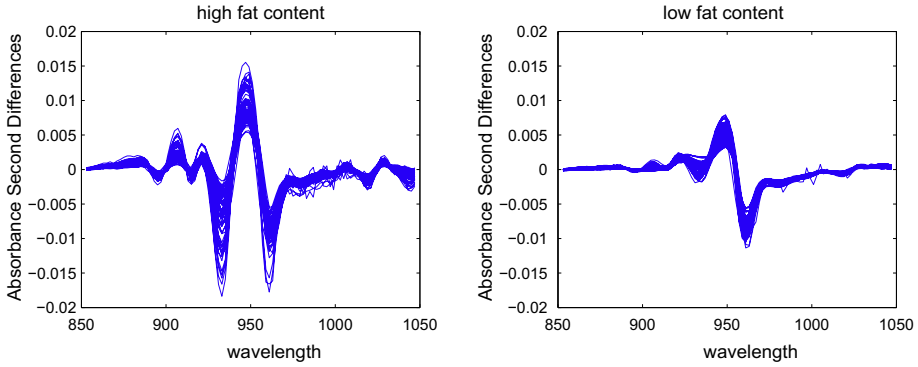

Fig. 2. Derivatives of the curves for `tecator` dataset.

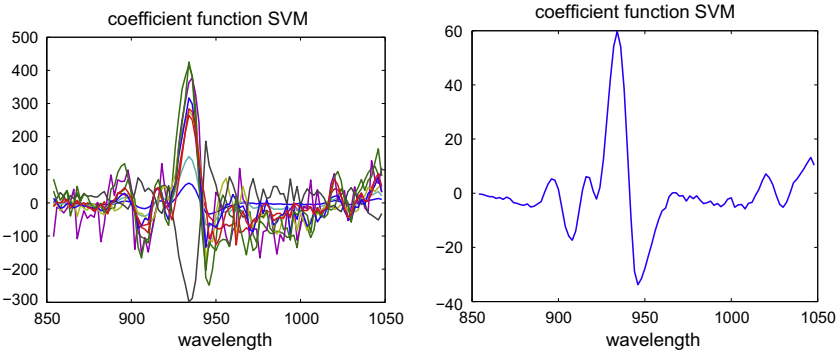

Fig. 3. Coefficient functions of SVM for `tecator` dataset.

but smooth quadratic-wise where it is non-zero. In this case, sparsity on both the zero and the third derivative is sought.

Let $\mathcal{D}$ be the set of the derivatives chosen to impose conditions. The proposed regularization term is $\sum_{d\in\mathcal{D}}\|w^{(d)}\|_1$, where $\|\cdot\|_1$ is the L_1 norm and $w^{(d)}$ is the d derivative of w or an approximation of it. This yields to the following optimization problem,

$$min_{w\in\mathcal{X},\beta\in\mathbb{R}}\sum_{d\in\mathcal{D}}\|w^{(d)}\|_1 + C\sum_{u=1}^{n}h(y_u, < w, X_u > +\beta). \tag{3}$$

The set of functions $\mathcal{X}$ can be a wide space, such as L^2, for which the optimization problem given in (3) becomes infinite dimensional. This issue is addressed in the next section via the use of a basis.

Note that when several derivatives are included in $\mathcal{D}$, it might also be convenient to give different weights to the different derivatives. We do not explore such issue, but it is a straightforward modification of (3).

Table 1
Classification accuracy in `tecator` database.

$\mathcal{D}$	Interpretation effect	Error
0	Sparse	1.0821**
0 and 1	Sparse and constant-wise	1.2800*
0 and 2	Sparse and linear-wise	1.2968*
0 and 3	Sparse and quadratic-wise	1.3558*
1	Constant-wise	1.5368*
2	Linear-wise	1.8232
3	Quadratic-wise	2.1600
Linear FSVM	None	3.28
Gaussian FSVM	None	2.6
Linear SVM	None	1.8779
FSDA	Detection of segments	1.09
RKHS	None	1.54

* Significantly better (t-test) than SVM.
** Significantly better (t-test) than all the others.

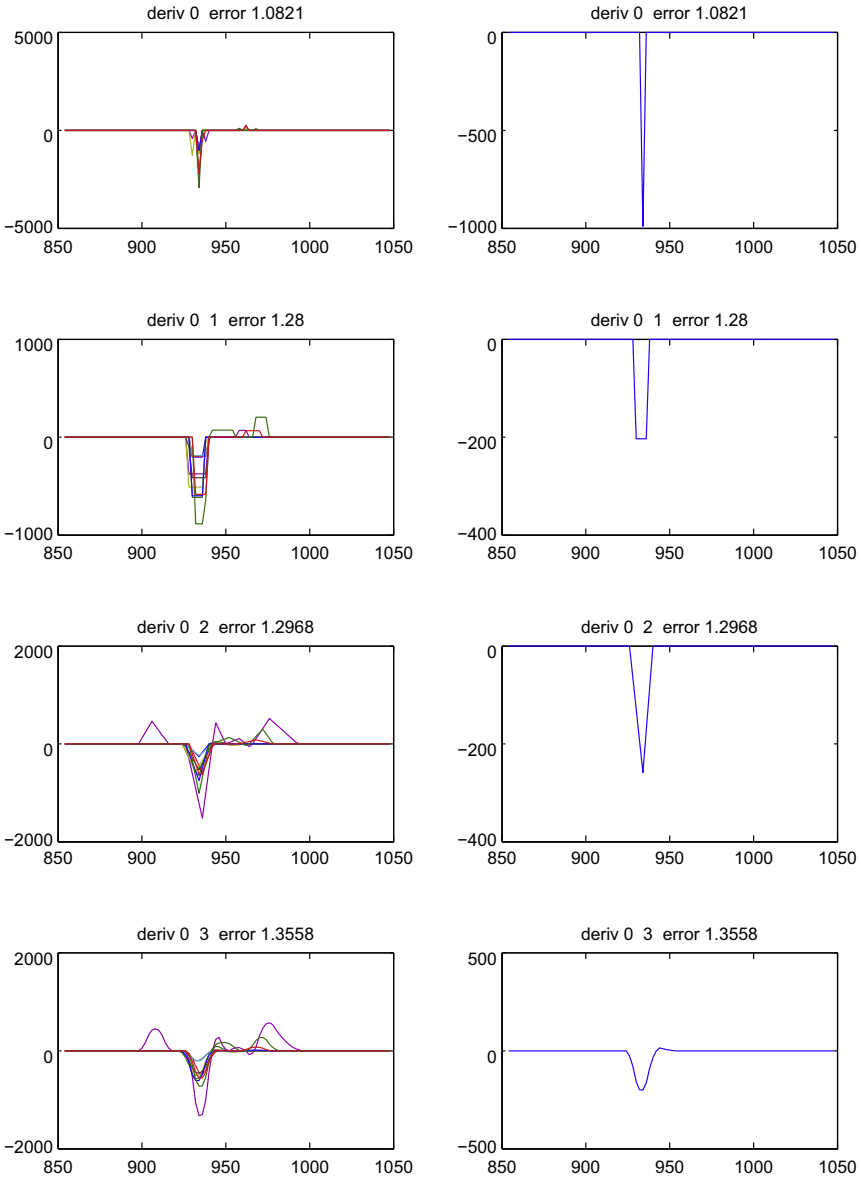

Fig. 4. Coefficient functions of ISVMFD for `tecator` dataset. Part I.

3.2. Implementation through the use of a basis

We consider the selection of a p-dimensional basis $B(t) = [b_1(t), b_2(t), \ldots, b_p(t)]^\top$, in such way that:

$$w(t) = B(t)^\top \eta. \tag{4}$$

Usually, p is assumed to be low in order to provide some form of regularization that avoids overfitting. However we work with p large enough to allow a perfect fitting. In our method, regularization is not based on the low dimension of B, but it is intrinsically related to the interpretability issue, since it is done by minimizing the L_1 norm of one or several derivatives of the score function w.

Our method can be applied to any high dimensional basis, such as splines or wavelets. Once we have a basis B, the score function can be rephrased as:

$$f(X_u) = \eta^\top x_u + \beta, \tag{5}$$

where $x_u = \int_{\mathcal{I}} X_u(t) B(t) dt$.

Note that it is not necessary to assume the basis functions $B(t)$ to be differentiable. Based on the choices of the practitioner, we are seeking a score function w that is sparse, constant-wise, linear-wise, quadratic-wise, etc. We propose to approximate the derivatives of $w(t)$ by its finite differences. Let $s_0, s_1, \ldots, s_r$ be a fine grid of the interval $\mathcal{I}$. Let $D^0 w = (w(s_0), w(s_1), \ldots, w(s_r))^\top$ be the discretization of the coefficient function w on such grid. An approximation of the d derivative of w can be obtained by the finite difference operator

$$D^d w(s_j) = \frac{D^{d-1} w(s_j) - D^{d-1} w(s_{j-1})}{s_j - s_{j-1}} \quad \text{for } j = 0, 1, \ldots, r - d. \tag{6}$$

Enforcing sparsity on $D^d w = (D^d w(s_0), D^d w(s_1), \ldots, D^d w(s_{r-d}))^\top$ yields a coefficient function w whose dth derivative is zero in all but a few points s.

Let $A_d = [D^d B(s_0), D^d B(s_1), D^d B(s_2), \ldots, D^d\ B(s_{r-d})]^\top$, where D^d is the finite difference operator defined in (6). Then, $\gamma = A_d \eta = D^d w$ is a good approximation of $w^{(d)}$ and hence, enforcing sparsity in γ pushes $w^{(d)}$ to be zero at most points t.

With this setting, (3) reduces to the vector optimization problem

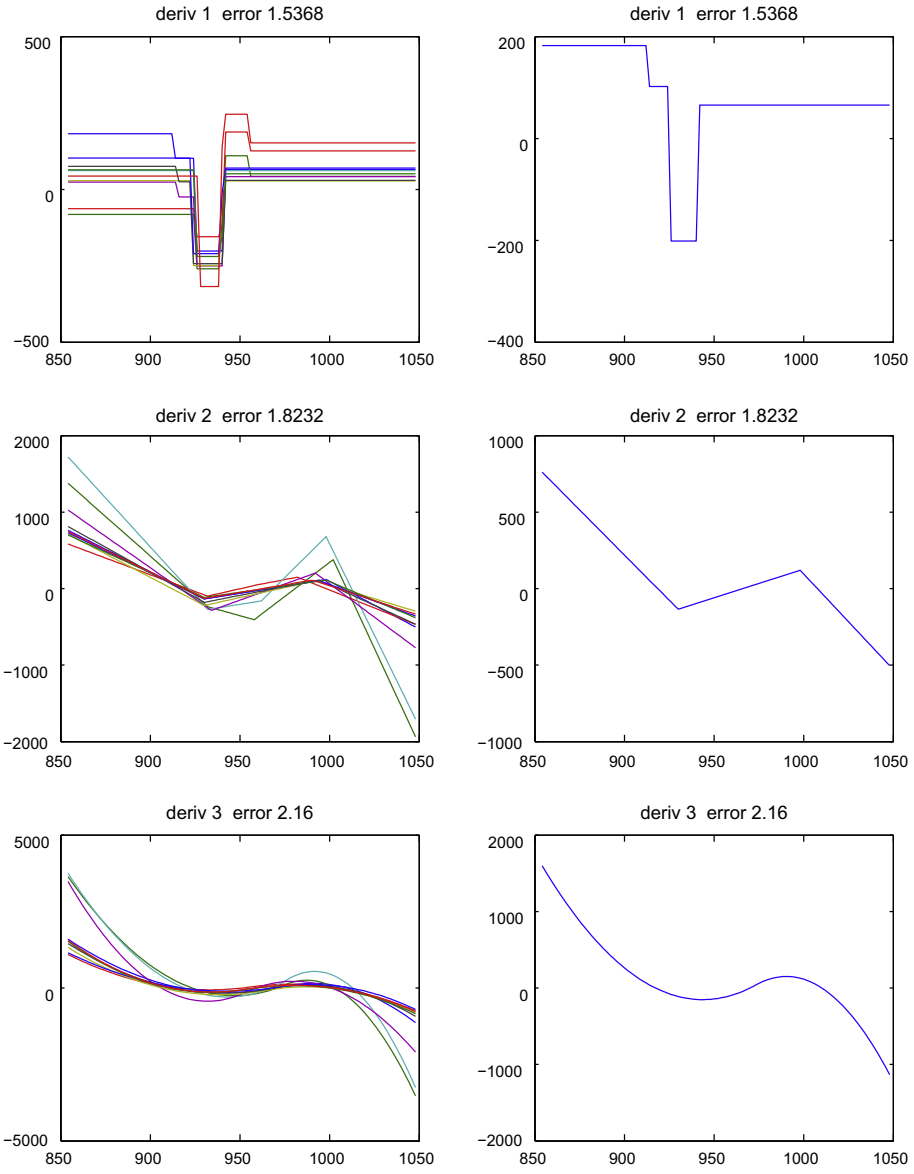

Fig. 5. Coefficient functions of ISVMFD for `tecator` dataset. Part II.

$$min_{\eta,\beta} \sum_{d\in\mathcal{D}} \|A_d\eta\|_1 + C\sum_{u=1}^{n} h(y_u, \eta^\top x_u + \beta), \tag{7}$$

which can be rephrased as the linear program

$$\begin{aligned} \min \quad & \sum_{d\in\mathcal{D}} e_{r+1-d}^\top z_d + C\sum_{u=1}^{n}\xi_u \\ \text{s.t.} \quad & y_u(x_u^\top\eta+\beta)+\xi_u \geqslant 1, \quad u=1,2,\ldots,n, \\ & -z_d \leqslant A_d\eta \leqslant z_d, \qquad d\in\mathcal{D}, \\ & \xi_u \geqslant 0, \qquad u=1,2,\ldots,n, \\ & z_d \in \mathbb{R}^{r+1-d}, \qquad d\in\mathcal{D}, \\ & \eta\in\mathbb{R}^{p+1}, \\ & \beta\in\mathbb{R}, \end{aligned} \tag{8}$$

where e_i is the i-dimensional vector with value one at each component.

As an example, consider the choice of a simple grid basis,

$$b_i(t) = \begin{cases} 1 & \text{if } t\in[t_{i-1}, t_i] \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

for all $i = 1,\ldots,p$. The grid used here does not necessarily coincide with the grid used in (6), although this is the option used in our numerical experiments. Note also that this grid is not differentiable, a condition that is not needed since our approach is based on finite differences. For this particular example, suppose that each function X_u is defined on the interval $\mathcal{I} = [0, p]$ and the grid $(0, 1, 2, \ldots, p)$ is considered both in (9) and (6). It can be easily seen that, $\eta = (w(0), w(1), \ldots, w(p))^\top$, A_0 is the identity matrix and $A_d = A_1^\top A_{d-1}$ for $d = 2,\ldots,p$. For example, A_1 and A_2 are equal to:

$$A_1 = \begin{pmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & -1 \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1 & -2 & 1 & \ldots & 0 \\ 0 & 1 & -2 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & -2 \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}. \tag{10}$$

The advantages of using finite differences to approximate the derivative of ω are twofold. First, the finite differences are useful when the basis functions are not differentiable, as it is the case of

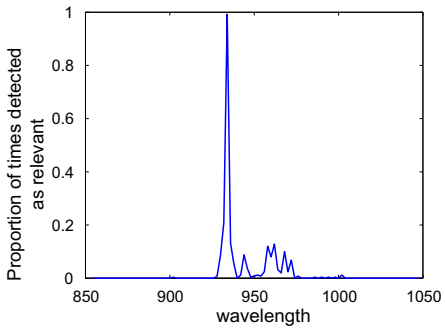

Fig. 6. Proportion of times that each channel is detected as important by the classifier `tecator` dataset.

the grid basis given in (9) and Haar wavelets. Secondly, they allow to transform the optimization problem (3) into a vector optimization problem, which indeed can be rephrase as a Linear Problem.

Alternatively, when differentiable basis functions are used, one might prefer to use the following expression of the derivative of ω

$$\omega^{(d)}(t) = \sum_{i=1}^{p} \eta_i b_i^{(d)}(t) = B^{(d)}(t)^\top \eta$$

where $b_i^{(d)}$ denotes the d derivative of the basis function b_i. In this case, in order to transform the problem into a Linear Problem, we still need a grid $s_0, s_1, \ldots, s_r$ to approximate $\|\omega^{(d)}\|_1$ by the L_1 norm of the vector

$$(B^{(d)}(s_0)^\top \eta, B^{(d)}(s_2)^\top \eta, \ldots, B^{(d)}(s_r)^\top \eta).$$

This approximation of $\|\omega^{(d)}\|_1$ can be used in problem (7) instead of $\|A_d \eta\|_1$ and the overall procedure does not change. Since in the illustrative examples in Section 5 nondifferentiable basis functions are considered, the former version that uses finite differences is the only option.

4. ISVMFD as a global framework for several existing methods

In this section we study how ISVMFD can be seen as a generalization of other methods available in the literature. In particular, L_1-norm SVM [4,6,7,29] and Fused SVM [36,33] turn out to be particular cases of ISVMFD for particular choices of the derivatives.

For linear SVM applied to vectors instead of functions, the L_1-norm SVM is a modification of SVM where the quadratic penalty term is replaced by the L_1-norm penalty of the coefficient vector. See for instance [4,40].

To simplify notation suppose that $\mathcal{I} = [0,1]$. Let $t_i = i/p$, for $i = 1, 2, \ldots, p$ be a regular grid on $[0,1]$. Suppose the functional datum X_u is known only on such grid. Consider that ISVMFD is used to select several time points. This means that the set of derivatives $\mathcal{D}$ in (3) should be set to $\{0\}$. We can represent the coefficient function w using a grid basis as in (9). Since X_u is unknown in the open interval (t_{i-1}, t_i), we consider $\frac{1}{p}X(t_i)$ as an approximation of

$$\int_{\mathcal{I}} X(t) b_i(t) dt = \int_{t_{i-1}}^{t_i} X(t) dt.$$

With this setting, application of ISVMFD to functions $\{X_u\}_{u=1}^n$ reduces to solving (7) with

$$x_u = \frac{1}{p}(X_u(t_1), X_u(t_2), \ldots, X_u(t_p))^\top, \quad \text{for all } u = 1, 2, \ldots, n.$$

In L_1-norm SVM, the L_1-norm penalty is known to act as a feature selection problem because it enforces the coefficient vector to be sparse. Hence, the L_1-norm SVM is able to produce a classifier that detects the several time points that are more relevant for classification. L_1-norm SVM applied directly to the vectors $\{x_u\}_{u=1}^n$ reduces to solving the following problem:

$$min_{\omega \in \mathbb{R}^p, \beta \in \mathbb{R}} \|\omega\|_1 + C \sum_{u=1}^{n} h(y_u, \omega^\top \hat{x}_u + \beta), \tag{11}$$

which is equivalent to (7).

Another method that can be seen as a particular case of ISVMFD is the Fused SVM. Fused SVM is the SVM-based counterpart of Fused Lasso, both proposed in [36]. Fused Lasso is a generalization of Lasso designed for problems whose features can be ordered in some meaningful way. It encourages both sparsity of the coefficient vector and sparsity of the differences between two consecutive components of the coefficient vector. Fused SVM seeks for a coefficient vector w that optimizes the following linear program:

$$\begin{aligned} \min \quad & \sum_{u=1}^{n} \xi_u \\ \text{s.t.} \quad & y_u\left(x_u^\top \eta + \beta\right) + \xi_u \geqslant 1, \quad u = 1, 2, \ldots, n, \\ & \sum_{j=1}^{p} |w_j| \leqslant s_1, \\ & \sum_{j=2}^{p} |w_j - w_{j-1}| \leqslant s_2, \\ & \xi_u \geqslant 0, \qquad u = 1, 2, \ldots, n, \\ & w \in \mathbb{R}^p, \\ & \beta \in \mathbb{R}, \end{aligned} \tag{12}$$

where s_1 and s_2 are two tuning parameters that trade off the loss term and the regularization terms (sparsity of w and sparsity of the differences). This problem is known to be equivalent to

$$\begin{aligned} \min \quad & \sum_{u=1}^{n} \xi_u + \lambda_1 \sum_{j=1}^{p} |w_j| + \lambda_2 \sum_{j=2}^{p} |w_j - w_{j-1}| \\ \text{s.t.} \quad & y_u\left(x_u^\top \eta + \beta\right) + \xi_u \geqslant 1, \quad u = 1, 2, \ldots, n, \\ & \xi_u \geqslant 0, \qquad u = 1, 2, \ldots, n, \\ & w \in \mathbb{R}^p, \\ & \beta \in \mathbb{R}, \end{aligned} \tag{13}$$

in the sense that, for any positive s_1 and s_2 on (12) there exist $\lambda_1, \lambda_2 > 0$, such that (η, β, ξ) is optimal for (12) if and only if it is optimal for (13).

Taking $\lambda_1 = \lambda_2$ and $C = \frac{1}{\lambda_1}$, (13) is the problem obtained when applying ISVMFD with $\mathcal{D} = \{0, 1\}$ and the grid basis (9).

5. Illustration on real databases

In this section we illustrate the usefulness of ISVMFD in two real publicly-available databases. First, we consider an application in spectrometry. The classification ability of ISVMFD is compared with other related methods that have previously used this database. The second example is an application in meteorology, where the aim is to discriminate the weather stations with a high-rain profile from the stations with a low-rain profile. The results for other three examples can be found in the Online companion Appendix.

5.1. Spectrometry data

The Tecator[1] data set consists of 215 near-infrared absorbance spectra of meat samples. These data are recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850–

[1] The data set is available at http://lib.stat.cmu.edu/datasets/tecator.

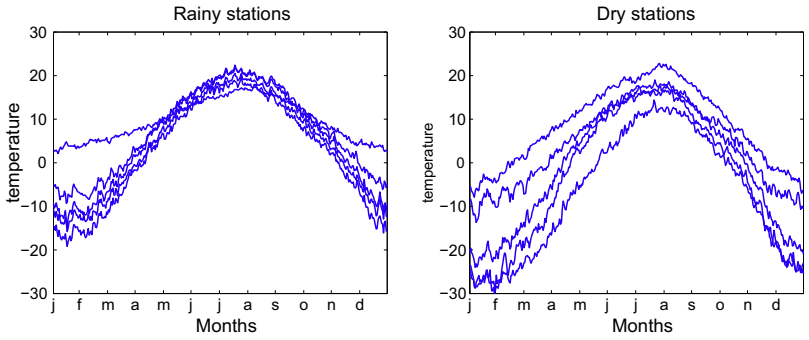

Fig. 7. Representation of data for `Rain` dataset.

Table 2
Classification accuracy in `rain` database.

$\mathcal{D}$	Interpretation effect	Error
0	Sparse	11.4286
0 and 1	Sparse and constant-wise	11.4286
0 and 2	Sparse and linear-wise	11.4286
0 and 3	Sparse and quadratic-wise	11.4286
1	Constant-wise	5.7143
2	Linear-wise	2.8571
3	Quadratic-wise	5.7143
SVM	None	8.5714

1050 nm by the Near Infrared Transmission (NIT) principle. Each sample contains finely chopped pure meat with different moisture, fat and protein contents. For each meat sample the data consists of a 100 channel spectrum of absorbances and the contents of moisture (water), fat and protein. The absorbance is $-\log 10$ of the transmittance measured by the spectrometer. The three contents, measured in percent, are determined by analytic chemistry.

Fig. 1 shows the spectra of the samples with high (left) and low (right) fat contents. The most important difference between these two sets of curves seems to be in their shape. High-fat curves tend to have two local minima whereas low-fat have only one. This suggests, as pointed out previously in [34], to use the second derivative of these curves instead of the original curves. Fig. 2 shows the curvature (second differences) of the curves.

For fair comparison with their results, we follow the same experimental setting as in [34]. Hence, we focus in the discrimination of samples with a low-fat content (less than 20%) versus high fat content (more than 20%). The dataset is split into 120 spectra for learning and 95 for testing. This splitting is repeated 250 times. For each splitting, the training set is again divided in two subsets: 60 spectra for learning and 60 spectra for validation. For each training set, the SVM is run in the learning set with the trade-off parameter C of SVM set to 10^i for $i = -1, 1, \ldots, 8$. The C with the best performance in the validation set is chosen and the SVM with such C is run again in the training set. Finally, the obtained classifier is evaluated in the testing set. This process is repeated 250 times and the average error on the testing set over the 250 repetitions is given. In all the experiments we use CPLEX 12.1 to solve the linear program (8). The whole algorithm is programmed in Matlab and it is available under request.

As suggested in Figs. 1 and 2, and in the empirical results obtained in [34,22], the second derivative of the spectra is more discriminative than the spectra itself. Hence, we focus on the use of such a second spectra. To approximate the second derivative, [34] uses a fixed spline subspace to represent the functions so as to calculate the second derivative. Instead of that, we apply the second finite difference operator D^2 defined in (6) to each function X_u. Classical linear SVM applied to this transformed data yields an error of 1.8779%, which is better than the results reported in [34] for FSVM (3.28% for the linear kernel and 2.6% for the Gaussian kernel). This example is also used in [11] where each functional datum is projected onto a Reproducing Kernel Hilbert Space (RKHS). Different kernels and different classifiers are tried. Among them, the best classification error reported is 1.54%.

In each practical application, the interpretability of the coefficient function issue might mean something different. For example, some practitioners might prefer to get a very sparse coefficient function, whereas others might prefer a linear-wise one. Different choices for the set of derivatives $\mathcal{D}$ yield different interpretation effects for the coefficient function. We have tried several sensible choices for these derivatives in order to compare them. Table 1 provides the interpretation effect and the classification error. The coefficient functions obtained for the first 10 runs are depicted in Figs. 4 and 5 left, the first of them is depicted on the right size to improve visualization.

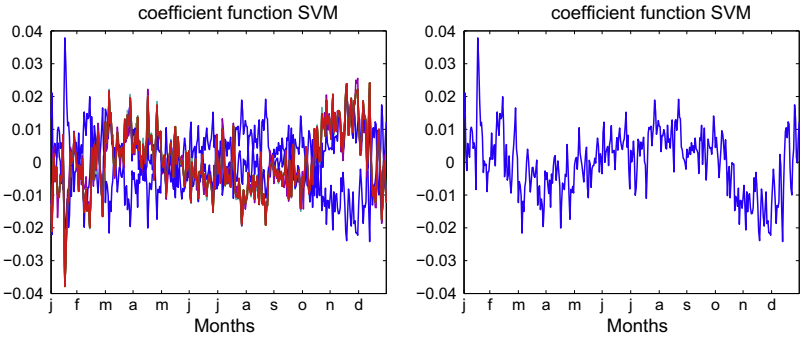

Fig. 8. Coefficient functions of SVM for `Rain` dataset.

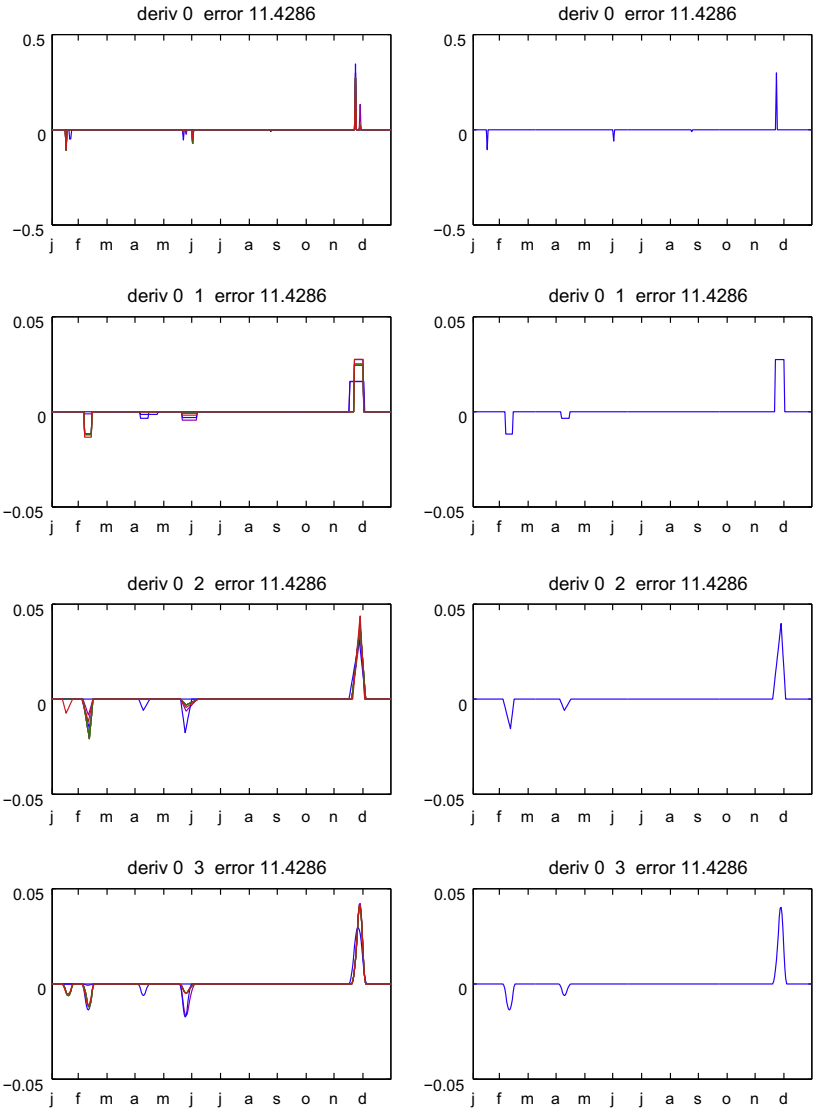

Fig. 9. Coefficient functions of ISVMFD for `Rain` dataset. Part I.

The best result in terms of classification performance is obtained for the sparse coefficient function. This error is very similar to the one provided in [22] (1.09%) by Functional Segment Discriminant Analysis (FSDA), a method that consists in a two-stage feature extraction followed by the application of SVM. We have not done a comparison with PDA, but [22] show that it performs worse than FSDA, whose error is similar to us.

Note that the horizontal axis of Fig. 2 represents the wavelength channel where the absorbance is measured. In this application, the detection of the channels with higher discriminative power is a key problem. Fig. 4 shows that direct application of ISVMFD clearly detects channel 935 as the most discriminative channel. Fig. 6 shows, for every channel, the relative frequency of being selected by ISVMFD over the 250 replications. It is clear that the channel 935 is selected almost always (99.2%), channels around it are also selected quite often and other channels are selected with a frequency bellow 15%. In [22] a similar experiment is reported for FSDA, with 50 replications, where the channel selected most frequently is also 935, but two other channels 905 ad 1045 are selected at remarkable frequencies too. Classical SVM, apart from getting worse classification ability, cannot be easily used to detect relevant channels as can be seen in Fig. 3 where the coefficient vector is shown.

5.2. Discriminating high-rain stations with weather data

The `weather` dataset consists of one year of daily temperature measurements from each 35 Canadian weather stations. Two experiments are conducted with this data, considering two different classification tasks: `regions` (Atlantic climate versus the rest) and `rain` (two classes are consider depending if the total yearly amount of precipitations are above or below 600). This experiment is inspired in the good interpretability results obtained in [17] for functional regression. We focus here in the `rain` example, the `regions` can be found in the Online Companion Appendix.

In this experiment, we follow a leave-one-out approach, where the training set is formed by all but one curve. This split is repeated for each curve in the dataset. Parameter C is chosen using half the training sample as learning set, and the rest for validation.

For the classification of `rain`, the original data can be seen in Fig. 7 and the results of ISVMFD for different choices of the interpretation effect can be seen in Table 2. The coefficient functions can be found in Figs. 9 and 10. In this case, contrary to the situation in previous example, we face a situation in which encouraging sparsity of $\|\omega\|$ yields, in general, worse results in terms of error rates. The best result is obtained for $\mathcal{D} = \{2\}$, which encourages piece-wise linear-

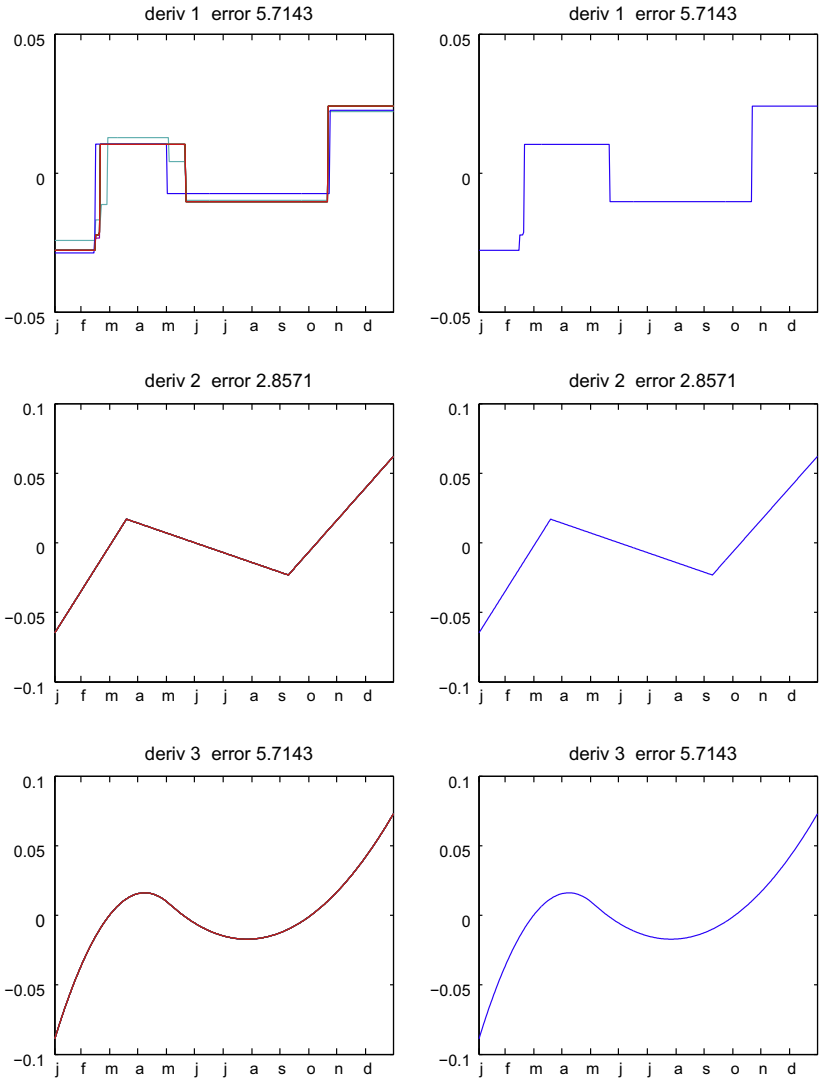

Fig. 10. Coefficient functions of ISVMFD for `Rain` dataset. Part II.

ity, without sparsity of the coefficient function itself. In Fig. 10, we see how the impact in favor of the positive class increases until mid March and then decreases until mid September, where it starts to increase again. As can be expected, the interpretation of the coefficient function obtained by SVM, shown in Fig. 8, is quite difficult.

6. Conclusions

In this paper we face the problem of obtaining an SVM-based classifier for functional data that has good classification ability and provides a classifier easy to interpret. The interpretability issue might strongly depend on the applications and the preferences of the user. Hence, we consider a flexible framework where different properties of the coefficient function are allowed. ISVMFD generalizes two other proposals available in the literature: the L_1-norm SVM and the Fused SVM. The experiments on real-world datasets show that ISVMFD produces an interpretable classifier that is competitive with SVM in terms of classification ability and similar in computational times.

Acknowledgements

The authors thank the anonymous referees and the associate editor for their helpful comments to improve the article. This work has been partially supported by projects MTM2009-14039, ECO2011-25706 of Ministerio de Ciencia e Innovación and FQM-329 of Junta de Andalucía, Spain.

References

[1] B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, Management Science 49 (3) (2003) 312–329.
[2] Bart Baesens, Christophe Mues, David Martens, Jan Vanthienen, 50 years of data mining and or: upcoming trends and challenges, Journal of the Operational Research Society 60 (1) (2009) S16–S23.
[3] N. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, International Journal of Computational Intelligence 2 (1) (2005) 59–62.
[4] P.S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines, in: Proceedings of the Fifteenth International Conferernce on Machine Learning (ICML98), Morgan Kaufmann, San Francisco, CA, 1998, pp. 82–90.
[5] E. Candes, T. Tao, The Dantzig selector: statistical estimation when p is much larger than n, The Annals of Statistics 35 (6) (2007) 2313–2351.
[6] E. Carrizosa, B. Martin-Barragan, D. Romero Morales, Binarized support vector machines, INFORMS Journal on Computing 22 (1) (2010) 154–167.
[7] E. Carrizosa, B. Martin-Barragan, D. Romero Morales, Detecting relevant variables and interactions in supervised classification, European Journal of Operational Research 213 (1) (2011) 260–269.
[8] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[9] F. Ferraty, P. Vieu, Curves discrimination: a nonparametric functional approach, Computational Statistics and Data Analysis 44 (1-2) (2003) 161–173.

[10] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer, 2006.
[11] J. González and A. Muñoz, Representing Functional Data in Reproducing Kernel Hilbert Spaces with Applications to Clustering and Classification. Technical Report 013, Statistics and Econometrics Series, 2010. ws102713.
[12] I. Guyon, A. Elisseff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[13] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh (Eds.), Feature Extraction, Foundations and Applications, Springer, 2006.
[14] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, The Annals of Statistics 23 (1) (1995) 73–102.
[15] T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer-Verlag, New York, 2001.
[16] R. Herbrich, Learning Kernel Classifiers, Theory and Algorithms, MIT Press, 2002.
[17] G.M. James, J.W. Wang, J. Zhu, Functional linear regression that's interpretable, The Annals of Statistics 37 (5) (2009) 2083–2108.
[18] C. Krier, F. Rossi, D. François, M. Verleysen, A data-driven functional projection approach for the selection of feature ranges in spectra with ICA or cluster analysis, Chemometrics and Intelligent Laboratory Systems 91 (1) (2008) 43–53.
[19] A. Laukaitis, Functional data analysis for cash flow and transactions intensity continuous-time prediction using Hilbert-valued autoregressive processes, European Journal of Operational Research 185 (3) (2008) 1607–1614.
[20] A. Laukaitis, A. Rackauskas, Functional data analysis for clients segmentation tasks, European Journal of Operational Research 163 (1) (2005) 210–216.
[21] Stefan Lessmann, Stefan Voß, A reference model for customer-centric data mining with support vector machines, European Journal of Operational Research 199 (2) (2009) 520–530.
[22] Bin Li, Qingzhao Yu, Classification of functional data: a segmentation approach, Computational Statistics and Data Analysis 52 (2008) 4790–4800.
[23] Jianping Li, Liwei Wei, Gang Li, Weixuan Xu, An evolution strategy-based multiple kernels multi-criteria programming approach: the case of credit decision making, Decision Support Systems 51 (2) (2011) 292–298.
[24] A.M. Lindquist, I.W. McKeague, Logistic regression with Brownian-like predictors, Journal of the American Statistical Association 104 (488) (2009) 1575–1585.
[25] D. Martens, B. Baesens, T. Van Gestel, Decompositional rule extraction from support vector machines by active learning, IEEE Transactions on Knowledge and Data Engineering 21 (2) (2009) 178–191.
[26] D. Martens, B. Baesens, T. Van Gestel, J. Vanthienen, Comprehensible credit scoring models using rule extraction from support vector machines, European Journal of Operational Research 183 (3) (2007) 1466–1476.
[27] J.M. Moguerza, A. Muñoz, Support vector machines with applications, Statistical Science 21 (3) (2006) 322–336.
[28] A. Muñoz, J. González, Representing functional data using support vector machines, Pattern Recognition Letters 31 (6) (2010) 511–516.
[29] J.P. Pedroso, N. Murata, Support vector machines with different norms: motivation, formulations and results, Pattern Recognition Letters 22 (2001) 1263–1272.
[30] J.O. Ramsay, C.J. Dalzell, Some tools for functional data analysis (with discussion), Journal of the Royal Statistical Society Series B 53 (3) (1991) 539–572.
[31] J.O. Ramsay, B.W. Silverman, Applied Functional Data Analysis, Springer-Verlag, New York, 2002.
[32] J.O. Ramsay, B.W. Silverman, Functional Data Analysis, Springer-Verlag, New York, 2005.
[33] F. Rapaport, E. Barillot, J.P. Vert, Classification of array CGH data using fused SVM, Bioinformatics 24 (13) (2008) 375–382.
[34] F. Rossi, N. Villa, Support vector machines for functional data classification, Neurocomputing 69 (7–9) (2006) 730–742.
[35] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society Series B 58 (1) (1996) 267–288.
[36] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji. Zhu, Keith Knight, Sparsity and smoothness via the fused lasso, Journal of the Royal Statistical Society Series B 67 (1) (2005) 91–108.
[37] Tony Van Gestel, David Martens, Bart Baesens, Daniel Feremans, Johan Huysmans, Jan Vanthienen, Forecasting and analyzing insurance companies' ratings, International Journal of Forecasting 23 (3) (2007) 513–529.
[38] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.
[39] Wouter Verbeke, Karel Dejaeger, David Martens, Joon Hur, Bart Baesens, New insights into churn prediction in the telecommunication sector: a profit driven data mining approach, European Journal of Operational Research 218 (1) (2012) 211–229.
[40] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-Norm support vector machines, Advances in Neural Information Processing Systems 16 (2003) 49–56.