Highlights

On using derivatives and multiple kernel methods for clustering and classifying functional data

Julien Ah-Pine, Anne-Françoise Yao

- Representing functional data in RKHS by means of kernel methods improves pattern recognition in both unsupervised and supervised learning settings.
- Combining functions with their derivative functions in a complementary manner can provide better clustering and classification performances
- Learning how to linearly combine functions with their derivative functions leads to more robust models.
- Our methods SF-MK-KM and SF-MK-SVM extend multiple kernel k-means and multiple kernel SVM to Sobolev functions.

On using derivatives and multiple kernel methods for clustering and classifying functional data

Julien Ah-Pine[a,b,c], Anne-Françoise Yao[b]

[a] *Université de Lyon, Lyon 2, ERIC UR 3083, 5 avenue Pierre Mendès-France, F69676, Bron Cedex, France*
[b] *Université Clermont Auvergne, LMBP UMR 6620, 3 place Vasarely, 63170, Aubière, France*
[c] *Université Clermont Auvergne, CERDI UMR 6587, CNRS, IRD, 26 Avenue Léon Blum, 63400, Clermont-Ferrand, France*

Abstract

In order to have a rich representation of functional data, we introduce a framework that relies on the following principles. Firstly, we pursue a multiview approach and consider the functions along with their derivative functions as distinct but complementary sources of information. Secondly, we assume that, in practice, functional data belong to non-linear manifolds and we thus promote kernel methods in order to cope with this hypothesis. Thirdly, we extend existing methods in multiple kernel learning for multivariate data to functional data. In this context, we present a general procedure that learns how to linearly combine the different kernel functions. We deal with the clustering and classification tasks. The methods that we introduce are extensions of the multiple kernel k-means and the multiple kernel SVM to Sobolev functions. Our experiments consider both simulated and real-world data and allow us to underline the advantages of our framework.

Keywords: Functional data analysis, Functional data clustering, Functional data classification, Derivative functions, Multiple kernel learning

1. Introduction

Our modern technologies allow one to massively record observations of diverse phenomena at fine grained resolutions in space and in time. For example, climate and environmental changes can be measured thanks to remote sensing instruments, machines health in facilities can be monitored

using sensors, human movements and physical activities, can be detected with a smartphone accelerometer sensor... These measurements are associated to a timestamp and/or a geographical location and are thus recorded as discrete data. But, they are in fact discretized observations of continuous curves or surfaces. From a data analysis standpoint, it is advantageoous to consider the continuous function underlying the multivariate data. Indeed, working with continuous functions allows one to leverage tools from functional analysis such as differential operators. Functional Data Analysis (FDA) is the branch of statistics that is concerned with this topic.

One main research line in FDA has been to extend multivariate statistical techniques and machine learning methods to functional data (FD). Concerning the clustering task, several works based on the k-means approach have been proposed in [1, 2, 3, 4, 5]. The Self-Organized Maps is yet another clustering technique that was studied in the context of FD [6]. Regarding the classification task, several machine learning models have also been adapted to FD. In that case, Rossi and colleagues for example, studied several machine learning methods: Radial Basis Function (RBF) Networks, Multi-Layer Perceptron (MLP) [7] and Support Vector Machines (SVM) as well [8].

In this paper, we propose to investigate the multiple kernel paradigm for clustering and classifying FD. Our motivations are the following ones. Firstly, similarly to the multivariate case, we argue that projecting FD onto Reproducing Kernel Hilbert Spaces (RKHS) can be beneficial in the non-linear case. Secondly, one interesting property of FD is that one can use the derivative functions so as to obtain a richer representation. More precisely, assuming that the FD belong to the Sobolev space $\mathbb{W}^{q,2}$, their successive derivatives up to order q, can provide one with q distinct sources of information. In this context, one can attempt to combine these different views in the goal of obtaining a better geometric representation of the FD in the context of clustering and classification tasks. When combining derivative functions, our main proposal is to employ the multiple kernel learning approach. For the clustering task, we suggest to extend the multiple kernel k-means approach to FD. As far as the classification problem is concerned, it is the multiple kernel SVM method that we aim to adapt for FD.

The rest of the paper is organized as follows. In section 2, we introduce the necessary background in FDA, make more precise the context we place ourselves in and discuss some related works on FD clustering, FD classification and on the use of derivatives in FDA. We also introduce an optimization procedure for learning how to balance the different views for both the un-

supervised and supervised cases. Then, in section 3, we review the kernel methods we focus on and introduce their extension to FD. In section 4, we report on the experimental results we obtained with the previous methods for clustering and classification tasks using both simulated and real-world datasets.

2. Notations, Background and Related work

In this section, we precise the general FDA setting we place ourselves into and introduce the required definitions and notations we use all along the paper. The different notions or work we respectively review concern: FD representation, FD clustering, FD classification and methods that combine derivative functions.

2.1. Functional data representation

We assume that the objects under study are smooth curves. More precisely, we suppose n real valued functions $\{x_i\}_{i=1,\dots,\mathrm{n}}$ in $\mathbb{W}^{\mathrm{q},2}([0,T]) \triangleq \mathbb{H}^{\mathrm{q}}([0,T])$ with $T > 0$, the Sobolev space that consists of functions x whose derivatives up to order q are elements of the Hilbert space $\mathbb{L}^2([0,T])$.

$$\mathbb{H}^{\mathrm{q}}([0,T]) = \{x \in \mathbb{L}^2([0,T]) : D^j x \in \mathbb{L}^2([0,T]), \forall j = 1,\dots,\mathrm{q}\} \qquad (1)$$

where D is the differential operator.

In practice, one does not directly observe the whole curves but samples of their realizations at different time points in $[0,T]$. While, the set of observation points of two distinct FD x_i and $x_{i'}$ can be different, say $\{t_{ij}\}$ and $\{t_{i'j'}\}$, we suppose, in this paper, that all FD were measured with respect to the same time grid $\{t_j\}_{j=1,\dots,\mathrm{p}}$.

In this case, a practical issue concerns the distribution of time points $\{t_j\}_j$ in $[0,T]$. The number of observations p can be large or restricted and the successive points $t_1 < t_2 < \ldots < t_{\mathrm{p}}$ can be equally spaced or not. In this contribution, we do not deal with such problems and place ourselves in the basic framework where the grid $\{t_j\}_j$ is assumed to be suitable so as to apply conventional pre-processing procedures in FDA (see for example [9]). Furthermore, we do not consider the registration problem of misaligned curves with respect to $\{t_j\}_j$ either.

Consequently, for all x_i, $i = 1,\dots,\mathrm{n}$, we suppose that we have a set of p observations $\{y_{ij}\}_{j=1,\dots,\mathrm{p}}$. However, we presume that these measurements

could have been corrupted by some noise:

$$y_{ij} = x_i(t_j) + \epsilon_{ij}, \quad \forall i, \forall j \tag{2}$$

where $\{\epsilon_{ij}\}_{i=1,\ldots,\mathrm{n};j=1\ldots,\mathrm{p}}$ are assumed to be independent across i and j.

From $\{y_{ij}\}_{i,j}$, one needs to reconstruct the functional form of the objects of interest, $\{x_i\}_i$. There are typically two ways to proceed: a data-driven method on the one hand, and using a set of basis functions on the other hand. The data-driven approach is based on functional principal component analysis (FPCA) and was initially introduced in [10]. However, since our proposal does not rely on this framework we do not review this method in what follows. The interested reader could refer to [9] for more information.

In this paper, we rather use a set of pre-defined basis functions. Several options can be considered in this case: Fourier functions, wavelets, B-splines... Depending on the kind of data, expert knowledge can help to select the most appropriate type of basis functions. In this contribution, we consider the commonly used B-splines basis system for its flexibility and properties that we exhibit subsequently. Since we assume that the derivatives up to the q^{th} order are in $\mathbb{L}^2([0,T])$, then we work with the subspace of functions spanned by the set of B-splines of order $\mathrm{d} = \mathrm{q} + 2$ so as to have a sufficiently rich framework to represent the functional data. Splines of order d are piecewise polynomial functions of order $\mathrm{d} - 1$. The domain $[0,T]$ of function x is split into several pieces $[t_1, t_2] \cup [t_2, t_3] \cup \ldots \cup [t_{\mathrm{p}-1}, t_{\mathrm{p}}]$. The function x is defined on each sub-interval by a local polynomial function of order $\mathrm{d} - 1$. In order to ensure continuity, two consecutive polynomials should be equal at their junction also called breakpoint. In a similar way, smoothness is attained by constraining the successive derivatives of two subsequent polynomials to be equal at their junction, up to order $\mathrm{d} - 2$. Given an order d and a set of p breakpoints $\{t_j\}_j$, the term B-splines refers to a unique set of spline functions that forms a basis systems for all spline functions of order d with breakpoints $\{t_j\}_j$. This basis systems is of dimension $\mathrm{m} = \mathrm{d} + \mathrm{p}$.

Consequently, let $\{\phi_k\}_{k=1,\ldots,\mathrm{m}}$ be a set of m B-splines that we also denote in vectorial form as $\boldsymbol{\phi} = (\phi_k)_{k=1,\ldots,\mathrm{m}}$. We assume that the FD are elements of the subspace $\mathrm{Span}(\phi_1, \ldots, \phi_{\mathrm{m}})$. In other words, $\forall i = 1, \ldots, \mathrm{n}$:

$$x_i = \sum_{k=1}^{\mathrm{m}} c_{i,k}\phi_k = \mathbf{c}_i^\top \boldsymbol{\phi}$$

where $\mathbf{c}_i$ is the $(\mathrm{m} \times 1)$ vector of coefficients of x_i in the B-splines basis system.

The smoothing step consists in estimating $\mathbf{c}_i$ given the observations $\{y_{ij}\}_{j=1,\dots,\mathrm{p}}$ for each element x_i in the sample. Because the measurements could have been corrupted by noise, one typically tackles the problem by a least square approach. However, the number of time points p could be very large, positioning the approximation problem in a high-dimensional framework. In order to avoid over-fitting and to have a better control over the smoothness of the FD, we penalize the sum of squared errors by a roughness penalty term denoted R. In this case, the lower $R(x_i)$ is, the smoother x_i will be.

More formally, the spline smoothing procedure amounts to solve:

$$\hat{\mathbf{c}}_i = \underset{\mathbf{c}\in\mathbb{R}^{\mathrm{m}}}{\arg\min} \sum_{j=1}^{\mathrm{p}} (y_{ij} - x_i(t_j))^2 + \lambda R(x_i) \tag{3}$$

where $x_i(t_j) = \sum_{k=1}^{\mathrm{m}} c_{i,k}\phi_k(t_j)$ and $\lambda > 0$ is a tuning parameter estimated by a cross-validation procedure which is based on the generalized cross-validation (GCV) criterion in our case.

It is important to note that using a pre-defined set of basis functions makes it possible to easily determine the successive derivative functions for each x_i. Indeed, thanks to the linearity of the differential operator, it is sufficient to determine the derivatives of the basis functions only.

2.2. Functional data clustering

Data clustering is the process that aims to partition a set of n elements into several groups called clusters, such that curves belonging to the same group are more similar to each other in comparison with other ones.

Clustering procedures are generally classified into two categories depending on the type of classification schemes they aim at (see for example [11]). Hierarchical clustering methods output a set of nested partitions that is encoded by a binary tree. In contrast, partitional clustering algorithms seek one partition according to an objective function. One can also make the distinction between hard and soft clustering methods depending whether an object exclusively belongs to one cluster or have non-null membership values with several clusters.

Many multivariate methods have been adapted or extended in order to cluster FD. Reviews of these approaches can be found in [12, 13].

In this contribution, we are interested in hard partitional clustering techniques. We focus on the k-means algorithm that aims to minimize the sum of squared errors and which is the most well-known method in this category.

In this context, we firstly quote [1] and [2] from the literature. In the former paper, the FD are projected onto a set of B-splines similarly to (3) whereas in the latter reference, the authors focus on Gaussian random functions and establish the link between the mean curves of each cluster found by k-means and eigenfunctions of the covariance function. In the work [3], a re-assignment procedure similar to k-means is carried out. However, each x_i is compared to its projection on the truncated Karhunen-Loève expansion of each current cluster. This paper is an instance of models that assume local linear functional subspaces associated to each cluster. Another interesting research work in this scope is [14]. The authors analyse the k-means problem in Hilbert spaces from a theoretical perspective. By extending the Johnson-Lindenstrauss lemma to separable Hilbert spaces, they propose yet another representation for FD based on random projections. In this latter framework, they provide an upper bound of the expected excess clustering risk. In the more recent paper [4], the k-means algorithm is also applied to FD. However, the FD representation is specifically based on a set of basis functions of an RKHS with a kernel function defined on $[0,T] \times [0,T]$. The smoothing procedure is carried out using a least-square approach with a Tikhonov regularization term following [15]. This projection of the FD on a RKHS leads to different strategies of dimension reduction (either by using the representer theorem or by the spectral representation based on the Mercer theorem). The FD are projected on reduced dimension subspaces and then the usual k-means is applied.

Differently from the latter paper, in our work, we project the FD from $\mathbb{L}_2([0,T])$ to another RKHS by using kernel functions defined on $\mathbb{L}_2([0,T]) \times \mathbb{L}_2([0,T])$ and leveraging the kernel trick. Similarly to vectorial data, we presume that FD can belong to non-linear subspaces and projecting them in another space can be beneficial. In the case of clustering, we employ the kernel k-means approach to reach that end. Several research works fall in this scope. In [4] a kernel k-means method is analyzed but, unlike our method, it is an approximate version which does not make use of all the data. Another related paper is [5]. In this work, while the k-means algorithm performs the partitioning, another variable is estimated at each iteration. It is a weight function defined on $[0,T]$ which is aimed at putting a null measure to the sub-intervals of $[0,T]$ that present very low variance. This weight function acts similarly to a feature selection procedure. The authors of [5] actually extend the framework defined in [16] from the multivariate to the functional case.

All previously cited research works apply the basic steps of the k-means algorithm: after an initialization, it re-assigns objects to their closest cluster, updates the clusters and their prototype then repeat these last two operations until convergence. In fact, the differences between these methods are mainly due to the kind of representation used for FD. In that respect, our contribution drift away from these works by assuming that the FD belong to a Sobolev space. This allows us to exploit the information coming from the derivative functions such as curvature. Moreover, we capitalize on the richness that RKHS allows through the application of kernel functions to FD. We elaborate further on this proposal in sub-section 3.2.

2.3. Functional data classification

In the case of classification tasks, we have at our disposal a set of n couples $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$ of $\mathbb{X} \times \mathbb{C}$ where $\mathbb{C}$ is a finite discrete set. $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$ is called a training set and it is employed in the goal of estimating a mapping $f : \mathbb{X} \to \mathbb{C}$ that aims to correctly predict $c \in \mathbb{C}$ for any given $x \in \mathbb{X}$.

There are many supervised learning frameworks in the literature. One can make the distinction between non-parametric and parametric approaches. In the non-parametric paradigm, there is no strong assumption on the form of the mapping f. This flexibility allows one to fit very precisely f on the training set. However, non-parametric techniques can suffer from over-fitting and finally provide bad predictions on unobserved objects. The interested reader can refer to [17] for a general treatment of non-parametric techniques in FDA.

In this paper, we deal with parametric models where the induction phase amounts to choose an appropriate instance from a class of functions, by minimizing a loss function. In this paradigm, there are numerous approaches. Several, classic multivariate methods have been extended to FD such as Linear Discriminant Analysis (LDA) [18], Quadratic Discriminant Analysis (QDA) [19] (see also the review paper [20]), logistic regression and more generally, generalized linear models [21, 22].

Predictive methods that stem from the machine learning community have also inspired researchers and practitioners dealing with FD. For instance, in [23], a functional random forest approach was introduced. Another random forest model for FD was also proposed in [24]. Neural networks and ensemble methods are other machine learning models that have also been examined in the case of FD such as in [7, 25, 26] and [27, 28] respectively.

In this paper, we focus on kernel methods. We firstly quote [29] which proposes to project FD in an RKHS for the regression and binary classification problems. In this latter paper, a penalized logistic loss function is employed. Next, in [30], the authors address the supervised learning task where the inputs are vectors of functions and the target output is a function as well. This framework is out of the scope of this paper since we study the more restricted case where the inputs are functions and the output is a discrete value. Nonetheless, it is worth noticing that [30] provides an interesting argumentation of the mutual benefits between FDA and machine learning.

The research work that constitutes a central ingredient of our proposal for supervised learning is detailed in [8] where the SVM model is discussed and extended in the context of FD. Let us assume the binary case where the target variable is in $\mathbb{C} = \{-1, 1\}$. The SVM method seeks an hyperplane represented by a linear functional $g : \mathbb{L}^2([0, T]) \rightarrow \mathbb{R}$ that separates the training set by taking into account two criteria. On the one hand, it maximizes the margin, that is to say, the distance between the hyperplane and the closest points. This criterion allows a better generalization of the resulting classifier. On the other hand, it minimizes the hinge loss, $\sum_i \max(0, 1 - c_i g(x_i))$, where $g(x_i)$ is related to the distance between x_i and the hyperplane defined by g. The hinge loss is more robust to outliers as compared to the squared loss. Important features of the SVM method are the following ones. Firstly, it leads to a convex optimization problem. Secondly, its dual formulation offers interesting flexibilities: it allows recasting the problem in terms of inner-products and this opens the gate to kernel functions and implicit non-linear extensions. In [8], the authors extend these principles for classifying FD. The functional nature of the data is discussed. In particular, transformations that are of interest for dealing with FD and which provide meaningful kernels are highlighted. We quote that projections of FD from an infinite dimensional Hilbert space to a finite subspace such as smoothing splines with a fixed subset of basis functions as described in sub-section 2.1 can be considered as such a transformation. Furthermore, a set of basis functions composed of splines makes it possible to easily determine the derivative functions. This feature is especially relevant in our case as already mentioned. Moreover, [8] establishes the consistency of functional SVM algorithm by adapting the framework introduced in [31] in the case of nearest neighbors. In our proposal, SVM for FD serves as a base tool in our supervised learning model for classifying FD in Sobolev spaces. In fact, our contribution, in the supervised case, can be viewed as a multiple kernel extension of [8].

2.4. Functional data analysis with derivatives

FD constitute rich objects to analyse. One specific feature is that one can work with the derivative functions which can encode additional discriminant information when it comes to clustering FD or learning a classifier from FD. In the domain of FDA, this was already pointed out at least since [32]. From a conceptual standpoint, semi-metrics derived from derivatives functions were emphasized by Ferraty and Vieu in [33, 17].

Regarding FD clustering, the research works described in [6, 17] show that, in the case of spectrometric data, the 2nd derivative functions can be more appropriate than the original functions. In the case of electrocardiograph curves, [34] shows that a composite distance measure that simply adds the distance between the original curves and the distance between the 1st derivative functions, leads to improved pattern recognition with the k-means algorithm. In [35] as well, it is exhibited that the k-means algorithm with a composite distance measure that sums up the distances between curves and between the derivative functions up to the 2nd order, can give better clustering performances. Both aforementioned papers apply a uniform weight when aggregating the pairwise proximity measures between the functions and the ones between the derivative functions. In contrast, the general framework introduced in [36], emphasizes the use of Sobolev metrics with non-uniform weights. However, the question of the estimation of the weights is left opened. In our clustering framework based on multiple kernel k-means, we integrate a step that estimates the influence of the derivative functions of distinct orders in the Sobolev metric.

Similarly, in the case of classification problems, quite a few research works have promoted the use of semi-metrics. The following reference, [37], is particularly relevant in this case. In the context of binary classification, the authors propose an LDA approach that learns how to combine discriminant features based on successive derivative functions. Other multivariate statistical methods extended to FD have also been examined with the integration of derivative functions. It is the case of the functional logistic regression (and more generally the generalized functional linear model) in [38], where derivative functions are included as functional covariates. Concerning machine learning techniques, we cite the approach introduced in [28] which is similar in spirit to [37]. In this work, several semi-metrics (*ie* derivative functions) with distinct types of distances are used to generate discriminant features.

Finally, another paper that studies derivative functions in supervised learning tasks is [39]. This approach builds upon [40] which define $\mathbb{H}^{\mathrm{q}}([0,T])$ as a direct sum of two reproducing kernel Hilbert subspaces, one being finite dimensional with a given set of basis functions and corresponding to the kernel of a given linear differential operator, and the other one being infinite dimensional and complementary to the previous one by means of boundary conditions. In this framework, [40] shows that the closed-form solution of the smoothing spline estimates of sampled FD can be formalized by a full rank linear operator from $\mathbb{R}^{\mathrm{p}}$ to $\mathbb{H}^{\mathrm{q}}([0,T])$. Then in [39], the authors use this solution as a pre-processing of the FD and eventually show that, in the framework described in [40], the subspace of $\mathbb{H}^{\mathrm{q}}([0,T])$ consisting of smoothing spline estimates equipped with the metric induced by the kernel function, is isomorphic to an Euclidean space equipped with a specific metric where usual multivariate techniques can then be carried out.

Unlike this latter setting, we are interested in individually considering the derivative functions of distinct orders up to the q^{th} one and in combining the information they convey by a weighting scheme. In addition, we propose to project the FD and their derivatives functions in several RKHS using the kernel trick. To our knowledge, these two points have not been jointly examined for both the clustering and the classification tasks.

3. Multiple kernel learning methods for clustering and classifying FD using derivatives

We now introduce in more details the different methods that we extend to FD. We assume that the FD are from the Sobolev space $\mathbb{W}^{\mathrm{q},2}$. Our purpose is to combine the information provided by the successive derivatives of the FD. Given the sample $\{x_i\}_i$ we can determine the derivative functions up to order q denoted $\{Dx_i\}_i, \{D^2x_i\}_i, \ldots, \{D^{\mathrm{q}}x_i\}_i$ which are interpreted as distinct views of the same objects. The different sets of functions can be implicitly mapped from $\mathbb{L}^2([0,T])$ to a RKHS using the kernel trick. Then we apply multiple kernel techniques in order to learn how to linearly combine the different single kernel functions. It should be clear that even though we suppose that the FD are elements of $\mathbb{W}^{\mathrm{q},2}$, we do not use the regular Sobolev metric:

$$\langle x_i, x_{i'} \rangle_{\mathbb{W}^{\mathrm{q},2}} = \sum_{s=0}^{\mathrm{q}} \langle D^s x_i, D^s x_{i'} \rangle_{\mathbb{L}^2}$$

where D^0 is the identity operator.

Instead, our contribution promotes non-uniform weights since we typically assume that the information conveyed by the derivatives are complementary to each other but some orders of derivation might be more important than other ones and should be more emphasized. Consequently, we assume a more general metric in $\mathbb{W}^{\mathrm{q},2}$:

$$\langle x_i, x_{i'} \rangle_{\mathbb{W}^{\mathrm{q},2}} = \sum_{s=0}^{\mathrm{q}} w_s \langle D^s x_i, D^s x_{i'} \rangle_{\mathbb{L}^2} \tag{4}$$

where $\mathbf{w} = (w_0, \ldots, w_{\mathrm{q}})$ and $w_s \geq 0, \forall s = 0, \ldots, \mathrm{q}$.

In this context, as far as the the clustering task is concerned, we learn how to weight each kernel function by looking at how they can further minimize the sum of squared errors given a partition. The underlying model is the multiple kernel k-means algorithm that we review in the case of multivariate data before its extension to our context.

Regarding the classification problem, we focus on the multiple kernel SVM approach. In the same spirit, in order to assign weights to the kernel functions associated to the derivative functions of different orders, we look at how the weights allow improving the SVM objective function.

Before introducing our extension of the multiple kernel k-means and the multiple kernel SVM for Sobolev functions in sub-sections 3.2 and 3.3, we introduce a common result to both models. It concerns the solution to the problem of learning the weight vector whose values are assigned to the different views given a separable cost function and under a ℓ_{r} norm constraint with $\mathrm{r} > 1$. It is noteworthy that we discard the ℓ_1 norm which provides sparse solutions. Indeed, our hypothesis is that the derivative functions of several orders are complementary to each other and our purpose is to design an aggregation scheme rather than a selection strategy.

3.1. Learning how to combine different views

In this sub-section, we assume that we are given a non-negative vector[1] $\mathbf{z} = (z_s)_{s=0,\ldots,\mathrm{q}}$ and we want to solve the following problem

$$\begin{aligned} &\max_{\mathbf{w} \in \mathbb{R}^{\mathrm{q}+1}} \mathbf{w}^\top \mathbf{z} \\ &\text{s.t. } \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1 \end{aligned} \tag{5}$$

[1]This vector will be properly defined in regard to the clustering and classification models we use, subsequently.

where $\mathbf{w} = (w_s)_{s=0,\dots,\mathrm{q}}$, $\mathbf{w} \geq \mathbf{0}$ is a shortcut for $w_s \geq 0, \forall s = 0, \dots, \mathrm{q}$ and $\|\mathbf{w}\|_{\ell_\mathrm{r}} = (\sum_s w_s^\mathrm{r})^{\frac{1}{\mathrm{r}}}$.

In our context, z_s reprensents the partial contribution of the s^{th} view $\{D^s x_i\}_{i=1,\dots,\mathrm{n}}$ to a given separable objective function, while the unknown w_s is the weight that indicates the importance that should be assigned to this latter view if one seeks to optimize $\mathbf{w}^\top \mathbf{z}$.

The weight vector should be non-negative and in order to bound the problem, we constrain the ℓ_r norm of $\mathbf{w}$ to not exceed 1. Problem (5) is convex and the following result establishes its closed-form solution.

Proposition 1. *Assuming* $\mathbf{z} \geq \mathbf{0}$ *and* $\mathrm{r} > 1$*, the solution to Problem* (5) *is given by,* $\forall s = 0, \dots, \mathrm{q}$*:*

$$w_s = \frac{z_s^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=1}^{\mathrm{q}} z_{s'}^{\frac{\mathrm{r}}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}}} \tag{6}$$

Proof. The Lagrangian function of Problem (5) reads:

$$L(\mathbf{w}, \boldsymbol{\alpha}, \beta) = \mathbf{w}^\top \mathbf{z} + \mathbf{w}^\top \boldsymbol{\alpha} + \beta(1 - \|\mathbf{w}\|_{\ell_\mathrm{r}})$$

where $\boldsymbol{\alpha} \in \mathbb{R}^\mathrm{n}$ and $\beta \in \mathbb{R}$ are the Lagrange multipliers which should be non-negative. Setting the derivative of L with respect to the primal variable to zero, it comes:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, \boldsymbol{\alpha}, \beta) = \mathbf{0} &\Leftrightarrow \mathbf{z} + \boldsymbol{\alpha} - \beta \frac{\mathbf{w}^{\mathrm{r}-1}}{\|\mathbf{w}\|_{\ell_r}^{\mathrm{r}-1}} = \mathbf{0} \\ &\Leftrightarrow \frac{\mathbf{w}^{\mathrm{r}-1}}{\|\mathbf{w}\|_{\ell_r}^{\mathrm{r}-1}} = \frac{\mathbf{z} + \boldsymbol{\alpha}}{\beta} \end{aligned}$$

where, by a slight abuse of notation, $\mathbf{w}^{\mathrm{r}-1} = (w_s^{\mathrm{r}-1})_{s=0,\dots,\mathrm{q}}$.
Clearly, β should be strictly greater than 0 and by the complementary conditions of the KKT conditions, this implies $\|\mathbf{w}\|_{\ell_\mathrm{r}} = 1$. Consequently, the previous equation simplifies into:

$$\mathbf{w}^{\mathrm{r}-1} = \frac{\mathbf{z} + \boldsymbol{\alpha}}{\beta}, \text{ that is to say, } w_s^{\mathrm{r}-1} = \frac{z_s + \alpha_s}{\beta}, \forall s = 0, \dots, \mathrm{q}$$

By hypothesis $z_s \geq 0$ and $\mathrm{r} > 1$. This implies $w_s \geq 0$ and thus, by the complementary conditions of the KKT conditions again, we deduce that $\alpha_s =$

$0, \forall s = 0, \ldots, \mathrm{q}$. From this reasoning, we obtain:

$$\mathbf{w} = \frac{\mathbf{z}^{\frac{1}{\mathrm{r}-1}}}{\beta^{\frac{1}{\mathrm{r}-1}}}, \text{ that is to say, } w_s = \frac{z_s^{\frac{1}{\mathrm{r}-1}}}{\beta^{\frac{1}{\mathrm{r}-1}}}, \forall s = 0, \ldots, \mathrm{q} \tag{7}$$

Now, using the activated constraint $\|\mathbf{w}\|_{\ell_\mathrm{r}} = 1$ again, it comes:

$$\begin{aligned} \left(\sum_s w_s^{\mathrm{r}}\right)^{\frac{1}{\mathrm{r}}} = 1 \Leftrightarrow \left(\sum_s \left(\frac{z_s}{\beta}\right)^{\frac{\mathrm{r}}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}} &= 1 \\ \Leftrightarrow \left(\sum_s z_s{}^{\frac{r}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}} &= \beta^{\frac{1}{\mathrm{r}-1}} \end{aligned} \tag{8}$$

Finally by plugging (8) into (7) we obtain the claimed solution. □

We further specify in what follows, how this result instantiates in the case of our clustering and classification models.

3.2. Multiple kernel k*-means for FD*

In the data mining community the idea to combine different views of the same set of elements in order to boost the performances of clustering methods dates back at least to [41]. The ever growing generation of objects presenting several representations such as web pages, annotated images or videos..., have encouraged the multi-view clustering research topic. A recent survey of methods is provided in [42]. In our case, we employ the multiple kernel k-means algorithm that was initially discussed in [43, 44] in the multivariate case. In [43], the estimation of the weights is computationally prohibitive unlike the closed-form solution that we emphasized in the previous sub-section. The authors of [44] introduced a so-called multiple view kernel k-means framework for multivariate data which also relies on an alternating optimization approach and on closed-form solutions. Nonetheless, their approach differs from our modeling and their solution is similar to that of the determination of the membership value of each object to each cluster in the fuzzy c-means method which was introduced in [45]. In this context, r is akin to the hyper-parameter which controls the partition fuzziness.

In our work, we propose to apply the multiple kernel k-means to FD in $\mathbb{W}^{\mathrm{q},2}$ where each set of derivative functions of order $s = 1, \ldots, \mathrm{q}$ is considered

as a distinct view. More formally, the multiple kernel k-means problem that we address can be casted as follows:

$$\min_{C,\mathbf{w}} \sum_{l=1}^{\mathrm{k}} \frac{1}{2|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s} \tag{9}$$

$$\text{s.t.} \begin{cases} C = \{C_1, \ldots, C_k\} \text{ is a partition,} \\ \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1. \end{cases}$$

where $\mathbb{F}^s$ is an RKHS onto which the functions $\{D^s x_i\}_i$ are projected by means of the mapping $\psi^s : \mathbb{L}^2 \to \mathbb{F}^s$.

The loss function is the within cluster variance which is a weighted mean of the within variance of each cluster C_l with $l = 1, \ldots, \mathrm{k}$. In order to establish a graph-based formulation that relies on kernel matrices, we express the within variance in a pairwise manner. We also make explicit the fact that the objective function is separable with respect to the different views $s = 0, \ldots, \mathrm{q}$.

Thanks to the decomposition of the total variance into the sum of the within and between clusters variances, we can maximize the between cluster variance instead and the previous problem is equivalent to the following one:

$$\begin{aligned} \max_{C,\mathbf{w}} \frac{1}{2\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s} \\ - \sum_{l=1}^{\mathrm{k}} \frac{1}{2|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s} \end{aligned} \tag{10}$$

$$\text{s.t.} \begin{cases} C = \{C_1, \ldots, C_\mathrm{k}\} \text{ is a partition,} \\ \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1. \end{cases}$$

Next, by expanding the squared distances in the previous objective function in terms of kernel functions $k^s(D^s x, D^s y) = \langle \psi^s(D^s x), \psi^s(D^s y) \rangle_{\mathbb{F}^s}$ and by denoting the evaluation of the latter expression for all pairs $\{(x_i, x_{i'})\}_{i,i'=1,\ldots,\mathrm{n}}$ of the sample by means of the square matrix $\mathbf{K}^s = (\mathbf{K}^s_{ii'})_{i,i'=1,\ldots,\mathrm{n}} = (k^s(D^s x_i, D^s x_{i'}))_{i,i'=1,\ldots,\mathrm{n}}$, it is not difficult to show that Problem (10) can be formulated as follows:

$$\max_{C,\mathbf{w}} \sum_{s=0}^{\mathrm{q}} w_s \left(\sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'} \right) \tag{11}$$

$$\text{s.t.} \begin{cases} C = \{C_1, \ldots, C_\mathrm{k}\} \text{ is a partition,} \\ \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1. \end{cases}$$

We apply the mainstream strategy for solving such kinds of multiple kernel learning problems which consists in alternating between maximizing with respect to C while keeping $\mathbf{w}$ fixed and then maximizing with respect to $\mathbf{w}$ while keeping C fixed. In the former case, a usual kernel k-means algorithm is employed to determine C. In the latter case, we have a closed-form solution following the materials we exposed previously. Let us introduce the following variable:

$$z_s = \sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'}, \forall s = 0, \ldots, \mathrm{q} \tag{12}$$

Note that $z_s \geq 0, \forall s = 0, \ldots, \mathrm{q}$. Then by applying Proposition 1, we get the following result.

Corollary 1. *Let $C = \{C_1, \ldots, C_{\mathrm{k}}\}$ be fixed and $\mathrm{r} > 1$, then the following optimization problem:*

$$\begin{aligned} &\max_{\mathbf{w}} \sum_{s=0}^{\mathrm{q}} w_s \left(\sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'} \right) \\ &s.t. \quad \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1. \end{aligned}$$

is convex and the optimal solution is given by, $\forall s = 0, \ldots, \mathrm{q}$:

$$w^*_s = \frac{\left(\sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'} \right)^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=1}^{\mathrm{q}} \left(\sum_{l=1}^{k} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^{s'}_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^{s'}_{ii'} \right)^{\frac{\mathrm{r}}{\mathrm{r}-1}} \right)^{\frac{1}{\mathrm{r}}}}$$

It is worth mentioning that the range of each kernel values k^s for $s = 1, \ldots, \mathrm{q}$ can strongly vary. Accordingly, before combining the different kernel matrices $\{\mathbf{K}^s\}_{s=1,\ldots,\mathrm{q}}$, it might be important to carry out a normalization procedure so as to make them more comparable to each other.

We denote our method by SF-MK-KM for Sobolev functions multiple kernel k-means and we wrap up its procedure in Algorithm 1.

Since the alternating procedure described in Algorithm 1 improves the objective function of Problem (11) at each iteration, it converges to a local optimum.

Algorithm 1: Sobolev functions multiple kernel k-means (SF-MK-KM).

Input: $\{y_{ij}\}_{i=1,\ldots,n;j=1,\ldots,\mathrm{p}}$ (sampled values of FD), $\mathrm{q} \geq 0$ (maximum order of derivative), $\mathrm{r} > 1$ (ℓ_{r} norm, default 2), $\{k^s\}_{s=0,\ldots,\mathrm{q}}$ (kernel functions, default Gaussian), σ (kernel hyper-parameter if any, default 1), $\mathrm{k} \geq 2$ (number of clusters)

Output: C (partition of FD), $\mathbf{w}$ (weight vector of size $\mathrm{q}+1$)

Project the sampled FD onto a pre-defined set of $\mathrm{q}+2+\mathrm{p}$ B-splines of order $\mathrm{q}+2$ and determine $\{x_i\}_{i=1,\ldots,\mathrm{n}}$ by solving (3);

Determine $\{D^s x_i\}_{i=1,\ldots,\mathrm{n}}, \forall s = 1, \ldots, \mathrm{q}$;

Determine $\{\mathbf{K}^s = (k^s(D^s x_i, D^s x_{i'}))_{i,i'=1,\ldots,\mathrm{n}}\}, \forall s = 0, \ldots, \mathrm{q}$;

Normalize the kernel matrices $\mathbf{K}^s, \forall s = 0, \ldots, \mathrm{q}$ (optional);

Initialize a uniform weight vector $\mathbf{w}$;

while *Stopping condition not reached* **do**

  Fix $\mathbf{w}$ and apply the kernel k-means algorithm with multiple kernel $\mathbf{K} = \sum_{s=0}^{\mathrm{q}} w_s \mathbf{K}^s$ to determine a new C (if applicable, use the previous C as for initialization);

  Fix C and apply Corollary 1 to determine a new $\mathbf{w}$;

end

3.3. Multiple kernel SVM for FD

We already provided in sub-section 2.3 some background regarding the extension of the SVM model to FD following the work introduced in [8]. We now give a more formal presentation of the SVM technique by recalling the dual optimization problem that makes it possible to employ kernel functions.

Given a training set $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$ the SVM approach consists in solving the following (primal) convex optimization problem:

$$\begin{aligned} &\min_{a_0\in\mathbb{R}, a\in\mathbb{L}^2} \frac{1}{2}\|a\|^2_{\mathbb{L}^2} + \mu\sum_{i=1}^{\mathrm{n}}\xi_i \\ &\text{s.t.} \begin{cases} c_i\left(a_0 + \langle a, x_i\rangle_{\mathbb{L}^2}\right) \geq 1-\xi_i, \forall i = 1,\ldots,\mathrm{n}; \\ \xi_i \geq 0, \forall i = 1,\ldots,\mathrm{n}. \end{cases} \end{aligned} \tag{13}$$

where $\mu \geq 0$ is a hyper-parameter that controls the balance between the soft-margin which is inversely proportional to $\|a\|^2_{\mathbb{L}^2}$ and the soft-error $\sum_{i=1}^{\mathrm{n}}\xi_i$.

The previous constrained optimization problem is equivalent to the following unconstrained problem:

$$\min_{a_0\in\mathbb{R}, a\in\mathbb{L}^2} \frac{1}{2}\|a\|^2_{\mathbb{L}^2} + \mu\sum_{i=1}^{\mathrm{n}}\max\left(0, 1-c_i\left(a_0+\langle a, x_i\rangle_{\mathbb{L}^2}\right)\right) \tag{14}$$

One major interest of the SVM methodology resides in its dual problem which is stated as follows:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathbb{R}^{\mathrm{n}}} \sum_{i=1}^{\mathrm{n}}\alpha_i - \sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}}\alpha_i\alpha_{i'}c_ic_{i'}\langle x_i, x_{i'}\rangle_{\mathbb{L}^2} \\ &\text{s.t.} \begin{cases} \sum_{i=1}^{\mathrm{n}}\alpha_i c_i = 0; \\ 0 \leq \alpha_i \leq \mu, \forall i = 1,\ldots,\mathrm{n}. \end{cases} \end{aligned} \tag{15}$$

The duality allows transforming the primal problem in an infinite dimensional space $\mathbb{L}^2$, into a dual problem in a finite dimensional space $\mathbb{R}^{\mathrm{n}}$. Furthermore, the dual solely depends on the inner-products between pairs of objects in the training sample. This feature makes it possible to implicitly project the FD in a RKHS thanks to the kernel trick. Let $\mathbf{K}$ be a square matrix of order n with general term $\mathbf{K}_{ii'} = \langle\psi(x_i), \psi(x_{i'})\rangle_{\mathbb{F}} = k(x_i, x_{i'})$, where $\mathbb{F}$ is a RKHS with reproducing kernel function k and ψ its associated feature map. Consequently, the general form of the SVM approach in its dual

expression is given by:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^{\mathrm{n}}} \sum_{i=1}^{\mathrm{n}} \alpha_i - \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \mathbf{K}_{ii'} \qquad (16)$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{\mathrm{n}} \alpha_i c_i = 0; \\ 0 \leq \alpha_i \leq \mu, \forall i = 1, \ldots, \mathrm{n}. \end{cases}$$

As mentioned previously in sub-section 2.3, the adaptation of the SVM technique to FD was already introduced in [8]. Our proposal is an extension to the multiple kernel framework in the goal of leveraging the functional nature of the objects we deal with. Assuming the FD belong to $\mathbb{W}^{\mathrm{q},2}$, we propose to employ a multiple kernel matrix:

$$\mathbf{K} = \sum_{s=0}^{q} w_s \mathbf{K}^s \qquad (17)$$

where $w_s \geq 0, \forall s = 1, \ldots, \mathrm{q}$, and for all couples $(x_i, x_{i'})$ in the training set, $\mathbf{K}^s_{ii'} = \langle \psi^s(D^s x_i), \psi^s(D^s x_{i'}) \rangle_{\mathbb{F}^s} = k^s(D^s x_i, D^s x_{i'})$ with the same definitions for ψ^s and $\mathbb{F}^s$ as given in sub-section 3.2.

Similarly to the unsupervised case, we exploit the fact that the derivative functions provide other views of the original objects and apply the multiple kernel learning paradigm to combine those distinct sources of information. Moreover, we project each set $\{D^s x_i\}_i$ for all $s = 0, \ldots, \mathrm{q}$, from $\mathbb{L}^2([0,T])$ to an RKHS by means of the mapping functions ψ^s. This feature is important when classes are not linearly separable. Overall, our framework provides a flexible representation of FD for classification purposes.

In the classification case, we also suppose that the kernel matrices $\{\mathbf{K}^s\}_{s=1,\ldots,\mathrm{q}}$ should be mixed in a complementary way instead of being in competition with each other. Therefore, we focus on the general approach studied in [46, 47] which highlights a ℓ_{r} regularization with $\mathrm{r} > 1$ by constraining $\|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1$. In fact, we extend the latter model from the multivariate case to the functional one and this amounts to solve the following problem:

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathrm{q}+1}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^{\mathrm{n}}} \sum_{i=1}^{\mathrm{n}} \alpha_i - \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \sum_{s=0}^{\mathrm{q}} w_s \mathbf{K}^s_{ii'} \qquad (18)$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{\mathrm{n}} \alpha_i c_i = 0; \\ 0 \leq \alpha_i \leq C, \forall i = 1, \ldots, \mathrm{n}; \\ \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1. \end{cases}$$

The optimization procedure is alike the unsupervised case and consists in alternating between maximizing with respect to $\boldsymbol{\alpha}$ with a fixed $\mathbf{w}$ (using the regular SVM algorithm) then minimizing according to $\mathbf{w}$ with a fixed α. The second problem has a closed-form solution that can be stated using Proposition 1 by considering the opposite of the minimization in $\mathbf{w}$, and with the following definition:

$$z_s = \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \mathbf{K}_{ii'}^s, \forall s = 0, \ldots, \mathrm{q} \tag{19}$$

Corollary 2. *Let $\boldsymbol{\alpha}$ be fixed and $\mathrm{r} > 1$, then the following optimization problem:*

$$\begin{aligned} &\min_{\mathbf{w} \in \mathbb{R}^{\mathrm{q}+1}} \sum_{i=1}^{\mathrm{n}} \alpha_i - \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \sum_{s=0}^{\mathrm{q}} w_s \mathbf{K}_{ii'}^s \\ &s.t. \quad \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1. \end{aligned}$$

is convex and the optimal solution is given by, $\forall s = 0, \ldots, \mathrm{q}$:

$$w_s^* = \frac{(\sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \mathbf{K}_{ii'}^s)^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=0}^{\mathrm{q}} (\sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \alpha_i \alpha_{i'} c_i c_{i'} \mathbf{K}_{ii'}^{s'})^{\frac{\mathrm{r}}{\mathrm{r}-1}} \right)^{\frac{1}{\mathrm{r}}}}$$

In Algorithm 2 we give the pseudo-code of our multiple kernel SVM procedure for Sobolev functions (SF-MK-SVM). Similarly to the clustering case, the overall objective function is improved at each iteration, therefore, Algorithm 2 converges to a local optimum.

4. Experiments

In this section we experiment with the models that we have introduced previously using simulated and real-world data. For both cases, we investigate the clustering and the classification tasks. Our theoretical framework supposes a flexible q, the upper bound of the derivation order we integrate in the representation. Nonetheless, we only examine the cases $\mathrm{q} = 1$ and $\mathrm{q} = 2$ in our empirical work. Likewise, even though the theoretical results given in Proposition 1 is valid for all $\mathrm{r} > 1$, we set $\mathrm{r} = 2$ in all the experiments we present subsequently. Besides, in all tests, the stopping condition in Algorithms 1 and 2 is triggered if a precision of 10^{-5} is reached for the objective function, or a maximal number of 10 iterations is achieved.

Algorithm 2: Sobolev functions multiple kernel SVM (SF-MK-SVM).

Input: $\{y_{ij}\}_{i=1,\ldots,\mathrm{n};j=1,\ldots,p}$ (sampled values of FD), $\mathrm{q} \geq 0$ (maximum order of derivative), $\mathrm{r} > 1$ (ℓ_{r} norm, default 2), $\{k^s\}_{s=0,\ldots,\mathrm{q}}$ (kernel functions, default Gaussian), σ (kernel hyper-parameter if any)

Output: $\boldsymbol{\alpha}$ (support vectors's weight), $\mathbf{w}$ (weight vector of size $\mathrm{q}+1$)

Project the sampled FD onto a pre-defined set of $\mathrm{q}+2+\mathrm{p}$ B-splines of order $\mathrm{q}+2$ and determine $\{x_i\}_{i=1,\ldots,\mathrm{n}}$ by solving (3);

Determine $\{D^s x_i\}_{i=1,\ldots,\mathrm{n}}, \forall s = 1, \ldots, \mathrm{q}$;

Determine $\{\mathbf{K}^s = (k^s(D^s x_i, D^s x_{i'}))_{i,i'=1,\ldots,\mathrm{n}}\}, \forall s = 0, \ldots, \mathrm{q}$;

Normalize the kernel matrices $\mathbf{K}^s, \forall s = 0, \ldots, \mathrm{q}$ (optional);

Initialize a uniform weight vector $\mathbf{w}$;

while *Stopping condition not reached* **do**

  Fix $\mathbf{w}$ and apply the SVM algorithm with multiple kernel $\mathbf{K} = \sum_{s=0}^{\mathrm{q}} w_s \mathbf{K}^s$ to determine a new $\boldsymbol{\alpha}$;

  Fix $\boldsymbol{\alpha}$ and apply Corollary 2 to determine a new $\mathbf{w}$;

end

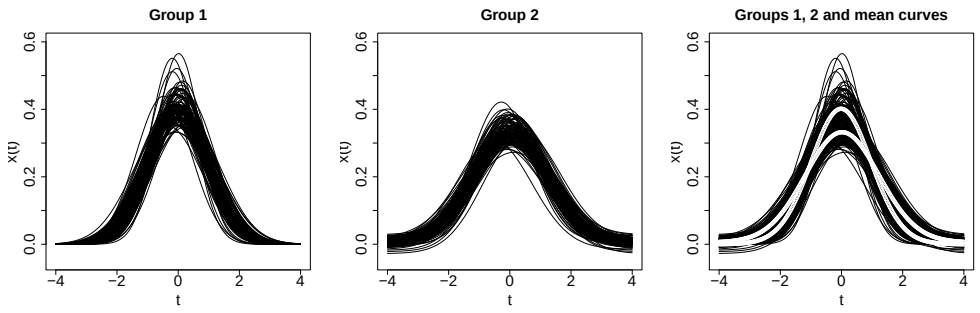

Figure 1: Gaussian functions in group 1 (left), in group 2 (middle), in both groups 1, 2 along with mean functions of group 1, 2 in white (right).

4.1. Simulated data

Firstly, we use a dataset of simulated Gaussian density functions on the domain $[-4, 4]$. We consider two groups 1 and 2, which are respectively associated to two distinct sets of parameters $(\mu_1, \sigma_1) = (0, 1)$ and $(\mu_2, \sigma_2) = (0, 2)$ along with the following random noises $\epsilon_{\mu_1}, \epsilon_{\mu_2} \sim \mathcal{N}(0, 0.15)$, $\epsilon_{\sigma_1} \sim \mathcal{N}(1, 0.1)$ and $\epsilon_{\sigma_2} \sim \mathcal{N}(1.2, 0.1)$. Furthermore, we add an extra source of variability for the second group by considering a random offset $\epsilon_a \sim \mathcal{N}(0.005, 0.01)$. Thereby, the first group of Gaussian curves is sampled as follows:

$$\begin{aligned} x(t) &= \mathcal{N}(t; \mu_1 + \epsilon_{\mu_1}, \sigma_1 + \epsilon_{\sigma_1}) \\ &= \frac{1}{(\sigma_1 + \epsilon_{\sigma_1})\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(t - (\mu_1 + \epsilon_{\mu_1}))^2}{(\sigma_1 + \epsilon_{\sigma_1})^2}\right), \end{aligned} \tag{20}$$

while the generative procedure for the Gaussian functions of group 2 is :

$$\begin{aligned} x(t) &= \mathcal{N}(t; \mu_2 + \epsilon_{\mu_2}, \sigma_2 + \epsilon_{\sigma_2}) + \epsilon_a \\ &= \frac{1}{(\sigma_2 + \epsilon_{\sigma_2})\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(t - (\mu_2 + \epsilon_{\mu_2}))^2}{(\sigma_2 + \epsilon_{\sigma_2})^2}\right) + \epsilon_a \end{aligned} \tag{21}$$

We represent 100 curves from each group 1 and 2 in Figure 1. It appears that making the distinction between the two sets is not completely straightforward since their bell shapes are pretty similar. However, in Figure 2, we respectively plotted the curves of the 1st and 2nd derivative functions (top and bottom rows respectively) of the two groups. We argue that using these

latter curves along with the original functions can help represent the FD more effectively for pattern recognition.

Indeed, in the graph on the right hand side of Figure 1, we plotted in white the Gaussian mean functions of the two groups and one can observe that they mainly differ around their common inflection point 0. Given the relationships between Gaussian density functions $\mathcal{N}(t;\mu,\sigma)$ with Hermite polynomials, it comes that the 1st and 2nd derivative functions have two and three inflection points which are $\{-\sigma,\sigma\}$ and $\{-\sqrt{3}\sigma,0,\sqrt{3}\sigma\}$ respectively. In Figure 2 we also observe that differences between the mean curves of the derivatives of the two groups are more pronounced around the aforementioned inflection points. In our perspective, this suggests that the derivative functions provide representations that can exhibit additional discriminative features that one could exploit.

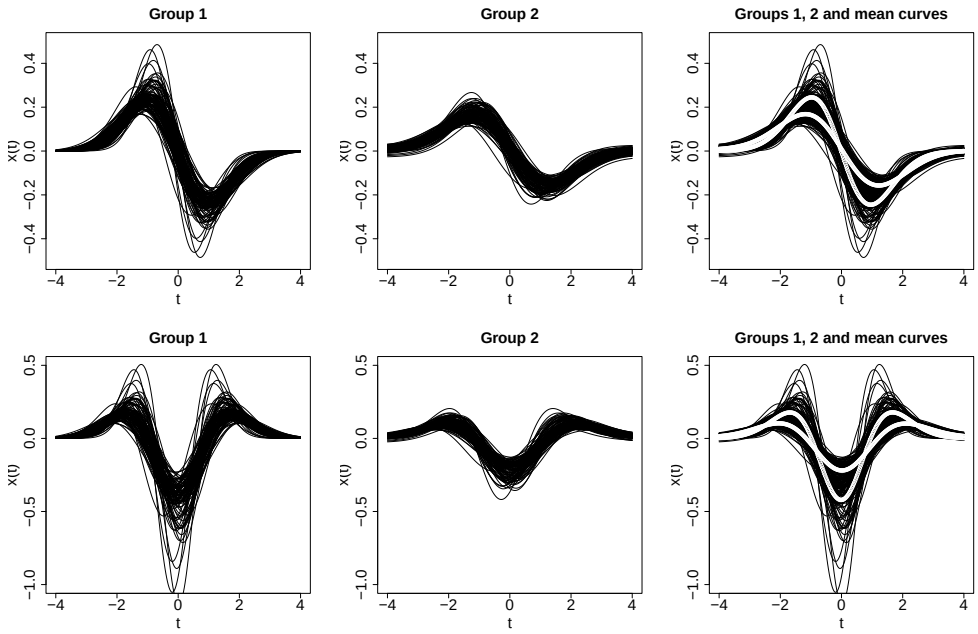

Figure 2: From top to bottom : 1st derivative functions (top row), 2nd derivative functions (bottom row). From left to right : curves in group 1 (left column), curves in group 2 (middle column), curves in both groups and mean functions of groups 1 and 2 in white (right column).

In what follows, our goal is to empirically study two research questions.

Firstly, can we improve pattern recognition of groups of functional data by by projecting them in RKHS ? This is related to the manifold hypothesis and our use of kernel methods : even though data are element of high dimensional spaces, in practice, it is often the case that they actually belong to non-linear manifolds with lower dimensions. Secondly, for functional clustering and classification tasks, is it beneficial to combine functions with their derivatives functions by using the multiple kernel techniques that we introduced previously ?

4.1.1. Clustering task

We start by studying the previous points in the case of clustering task. We take the previous sample of 100 functions of group 1 mixed with a 100 curves of group 2. Then, we alternatively applied the SF-MK-KM approach depicted in Algorithm 1 with $\mathrm{k} = 2$, using the following linear kernel matrices which are given acronyms for notational purposes :

- $00 : \mathbf{K}^{00} = (\langle x_i, x_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\ldots,\mathrm{n}}$,
- $11 : \mathbf{K}^{11} = (\langle Dx_i, Dx_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\ldots,\mathrm{n}}$,
- $22 : \mathbf{K}^{22} = (\langle D^2x_i, D^2x_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\ldots,\mathrm{n}}$,
- $01 : \mathbf{K}^{01} = \mathbf{K}^{00} + \mathbf{K}^{11}$,
- $02 : \mathbf{K}^{02} = \mathbf{K}^{00} + \mathbf{K}^{11} + \mathbf{K}^{22}$.

Note that the kernel matrices $\mathbf{K}^{01}$ and $\mathbf{K}^{02}$ are equivalent to using the regular Sobolev metric of $\mathbb{W}^{1,2}$ and $\mathbb{W}^{2,2}$, given in (3).

The clustering performances are assessed using external validation criteria where we compare the partition C obtained by the clustering method against the ground-truth that we denote by L. We use two conventional evaluation measures, the purity and the normalized mutual information. Let $L = \{L_1, \ldots, L_{\mathrm{k}}\}$ and $C = \{C_1, \ldots, C_{\mathrm{k}}\}$ denote the true classes and the found clusters respectively. Then, the Purity and the NMI assessment measures are defined by:

$$\mathrm{Purity}(C, L) = \frac{1}{\mathrm{n}} \sum_{l=1}^{\mathrm{k}} \max_{m=1,\ldots,\mathrm{k}} (|C_l \cap L_m|) \tag{22}$$

$$\mathrm{NMI}(C, L) = \frac{2\mathrm{MI}(C, L)}{\mathrm{H}(C) + \mathrm{L}(C)} \tag{23}$$

where $\mathrm{H}(C)$ is the entropy of C given by $\mathrm{H}(C) = -\sum_{l=1}^{\mathrm{k}}(|C_l|/\mathrm{n})\log((|C_l|/\mathrm{n}))$, and $\mathrm{MI}(C, L)$ is the mutual information between C and L expressed by $\mathrm{MI}(C, L) = \sum_{l,m=1}^{\mathrm{k}}(|C_l \cap L_m|/\mathrm{n})\log(\mathrm{n}|C_l \cap L_m|/(|C_l||L_m|))$.

Purity and NMI scores are in $[0, 1]$ and the higher the value is, the closer the two partitions are and the better the clustering solution is.

SF-MK-KM relies on the k-means heuristic and it is well-known that the random initialization of this clustering algorithm leads to different local optima. In order to address this source of variability, we ran SF-MK-KM 50 times on matrices $\mathbf{K}^{00}, \mathbf{K}^{11}, \mathbf{K}^{22}, \mathbf{K}^{01}$ and $\mathbf{K}^{02}$.

The box plots of the Purity and NMI measures obtained from our experiments are depicted in Figure 3. Note that we also provide the mean value, shown by a red triangle.

Let us first discuss the results obtained using the single kernel matrices $\mathbf{K}^{00}, \mathbf{K}^{11}, \mathbf{K}^{22}$. The three different views provide distinct performances. For the sample under study, the 1st derivative functions is ranked 1st, followed by the original functions and finally the 2nd derivative functions. If we uniformly add $\mathbf{K}^{00}$ to $\mathbf{K}^{11}$ to form the multiple kernel matrix $\mathbf{K}^{01}$, then the performances are comparable to $\mathbf{K}^{11}$ which are the two best overall representations. If we look at $\mathbf{K}^{02} = \mathbf{K}^{00} + \mathbf{K}^{11} + \mathbf{K}^{22}$, then it is a little bit "contaminated" by the lower assessment values of $\mathbf{K}^{22}$. Nonetheless, we argue that $\mathbf{K}^{02}$ allows combining the three single kernel matrices in a positive way in the sense that its evaluation scores are not lower than the minimum values of the single kernel matrices.

The next point that we address is the application of a kernel function in order to evaluate the manifold hypothesis on the simulated dataset. To this end, we use the Gaussian kernel which is given as follows for a couple of curves $(x_i, x_{i'})$:

$$\mathbf{K}_{ii'} = \exp\left(\frac{-\|x_i - x_{i'}\|^2_{\mathbb{L}_2}}{\sigma^2}\right) \tag{24}$$

where $\sigma > 0$ is an hyper-parameter controlling the neighborhood width.

This proximity measure was used for all single representations $\mathbf{K}^{00}$, $\mathbf{K}^{11}$, $\mathbf{K}^{22}$ and consequently for the multiple kernel matrices $\mathbf{K}^{01}$ and $\mathbf{K}^{02}$ as well.

After some preliminary tests, we found that $\sigma = 1$ was a good setting. Therefore, we use this value in all the experiments in this sub-section when using the Gaussian kernel. The clustering performances we obtained are exposed in Figure 4. By comparing the ranges of the values in the y axes of

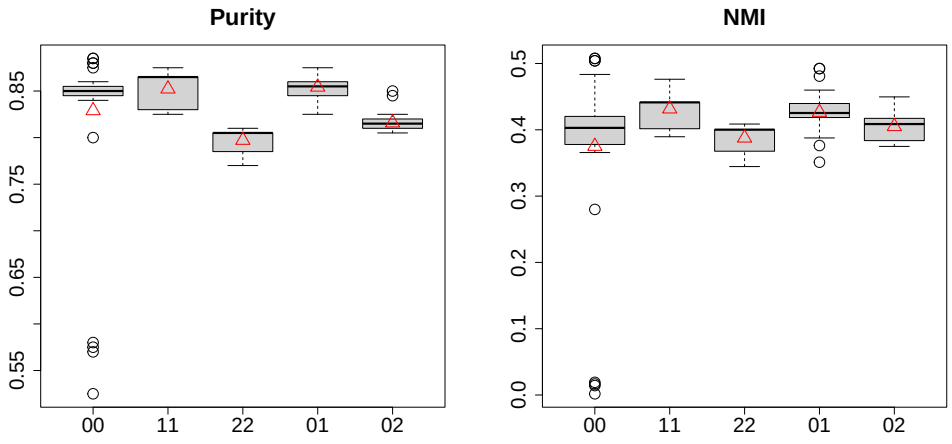

Figure 3: Box plots of Purity (left) and NMI (right) measures of SF-MK-KM without weight optimization and using linear kernels.

Figures 3 and 4, one can note that the Gaussian kernel outperforms the linear kernel for all representations except for 00. This shows that representing functional data in RKHS using kernel functions can lead to better clustering solutions.

Next, we investigate the weight optimization procedure when linearly combining the single kernel matrices. We introduce additional notations :

- 01o : $\mathbf{K}^{01o} = w_0\mathbf{K}^{00} + w_1\mathbf{K}^{11}$,
- 02o : $\mathbf{K}^{02o} = w_0\mathbf{K}^{00} + w_1\mathbf{K}^{11} + w_2\mathbf{K}^{22}$.

where $\mathbf{w} = (w_s)$ is the vector of positive weights that is updated at each iteration of Algorithm 1.

In Figure 6, we show the comparison between the Purity and the NMI values obtained with SF-MK-KM without weight optimization (02), and with weight optimization (02o). Clearly, optimizing the weights allows enhancing both evaluation criteria. Another interesting effect that is worth emphasizing is that it also reduces the variability caused by the k-means random initialization. Indeed, almost all 50 trials converge to the same performance measure for 02o.

In order to have a more global assessment of the SF-MK-KM approach, we generated 100 samples of the same kind of datasets as previously (100 curves

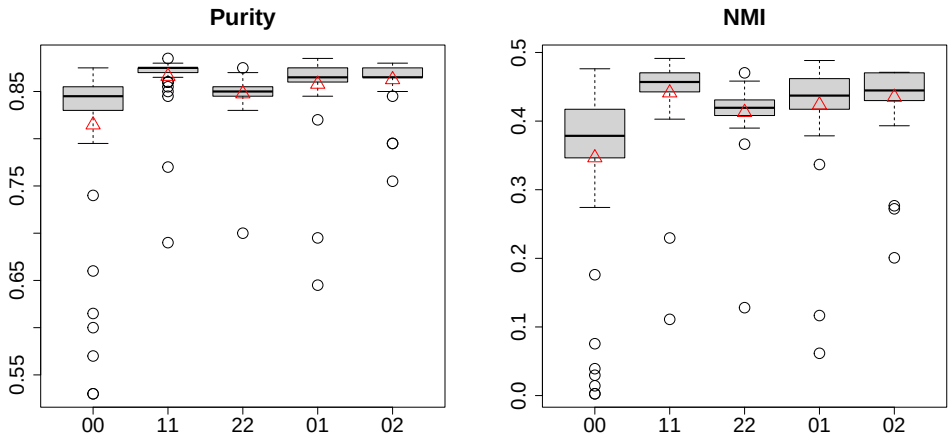

Figure 4: Box plots of Purity (left) and NMI (right) measures of SF-MK-KM without weight optimization and using the Gaussian kernel.

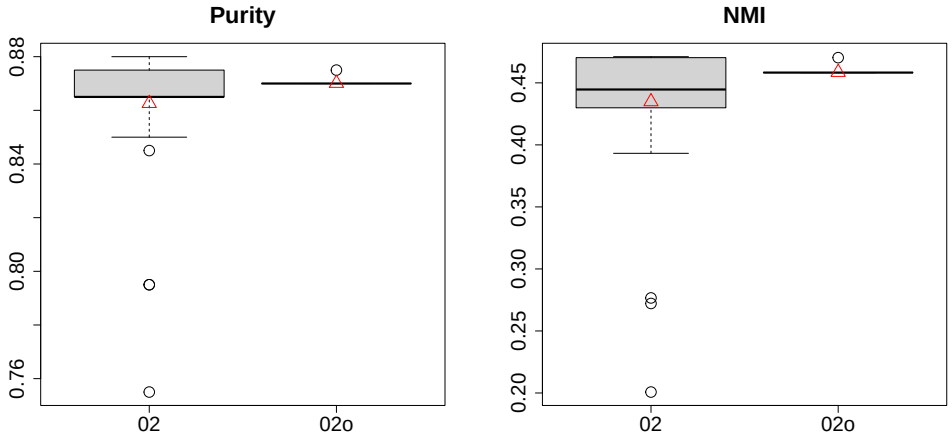

Figure 5: Box plots of Purity (left) and NMI (right) measures of SF-MK-KM without (02) and with (02o) weight optimization and using Gaussian kernels.

per group). We apply the same representations and parameters as before except for the number of random initializations. Indeed, for each sample, SF-MK-KM is run 10 times instead of 50. We averaged the performance

measures over these 10 trials.

In Figure 6, we plotted the box plots of the measures derived from the 100 samples for both the linear and Gaussian kernels whose results are indicated with prefix l and prefix g respectively. It allows us to confirm that the Gaussian kernel scores are superior to the linear kernel ones. The baseline representation 00 is somewhat an exception since both kernels give comparable results. Moreover, Figure 6 exhibits the positive effect of weight optimization since g01o outperforms g01 and so is the case when comparing g02o and g02. Overall, g02o (right most box plot in Figure 6) is the SF-MK-KM setting that provides the best results if we consider the mean values (red triangles).

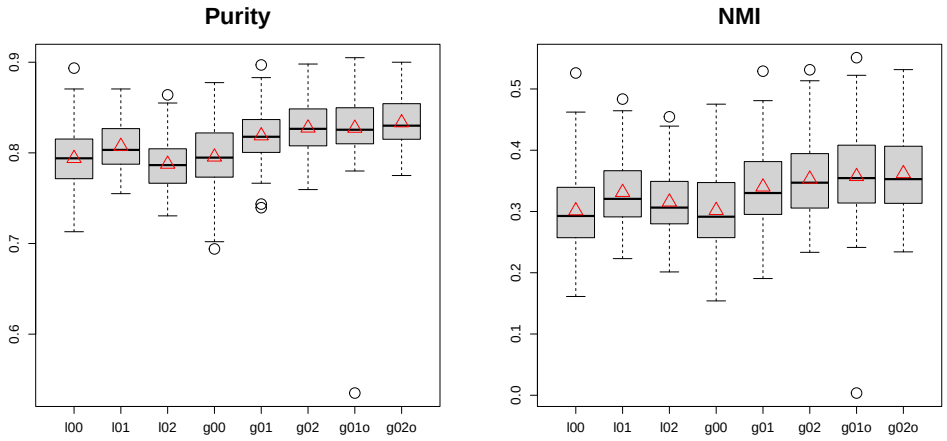

Figure 6: Box plots of averaged Purity (left) and averaged NMI (right) measures of SF-MK-KM tested on 100 samples of 200 curves (100 in each group) using linear and Gaussian kernels. The prefix l and g respectively stand for linear and Gaussian kernels. The suffix o stands for weights optimization.

4.1.2. Classification task

In this paragraph, we examine the classification task using the same simulated data as previously in order to evaluate the SF-MK-SVM model. The same representations 00, 11, 22, 01, 02, 01o, 02o, and both the linear and the Gaussian kernels were tested. Similarly to the clustering task, we set $\sigma = 1$ for the Gaussian kernel hyper-parameter. In the supervised learning context,

he evaluation criterion we used is the accuracy rate given by :

$$\text{Accuracy}(C, L) = \frac{1}{\text{n}} \sum_{l=1}^{\text{k}} |C_l \cap L_l| \tag{25}$$

where L is the true class distribution and C is the the one predicted by SF-MK-SVM.

In regard to the hyper-parameter μ in the SVM objective function, we tested the following values, $\mu \in \{0.1, 1, 10\}$, in a grid search fashion. In other words, we kept the best test error measure among the three alternatives. Furthermore, we applied a 10-fold cross-validation procedure.

The box plots in Figure 7 expose the accuracy dispersions among the 10 folds and the red triangles indicate the mean values.

The range of the y values between the two plots are different. Focusing on the median and mean accuracy scores, one can note that the measures on the left graph are lower than the ones on the right. Therefore, the Gaussian kernel provides better results in comparison to the linear kernel for the classification task as well. However, concerning the impact of combining different views of the curves based on their derivative functions, the observations are different from the clustering case. For the linear kernel (left graph in Figure 7), the results are negative : optimizing the weights hurt the performances. Our explanation for this phenomenon is that the single linear kernel matrices $\mathbf{K}^{00}$, $\mathbf{K}^{11}$ and $\mathbf{K}^{22}$, have heterogeneous ranges and caused a distortion in the importance of the different views in the combination. On the contrary, the Gaussian kernels are non-negative cosine measures whatever the representations. For the three single kernel matrices, their diagonal is 1. This implies that Gaussian kernel matrices are comparable to each other unlike the linear kernel matrices. As a consequence, the weight optimization in this case does not provide under-performances as depicted in the right graph of Figure 7. Nonetheless, it does not boost the accuracy measures neither. If we look at the median scores, they are all equal to 0.95 whatever the representation.

Likewise the clustering task, we tested our model more globally by applying it to 100 simulated samples in the same setting as described previously. The results are exhibited in Figure 8. It allows us to confirm that the Gaussian kernel gives better results than the linear kernel. However, the weight optimization did not provide any improvement and tend to degrade the performances a little bit. Still, we argue that the outcomes are pretty comparable among g00, g01, g02, g01o and g02o. Our explanation, in this

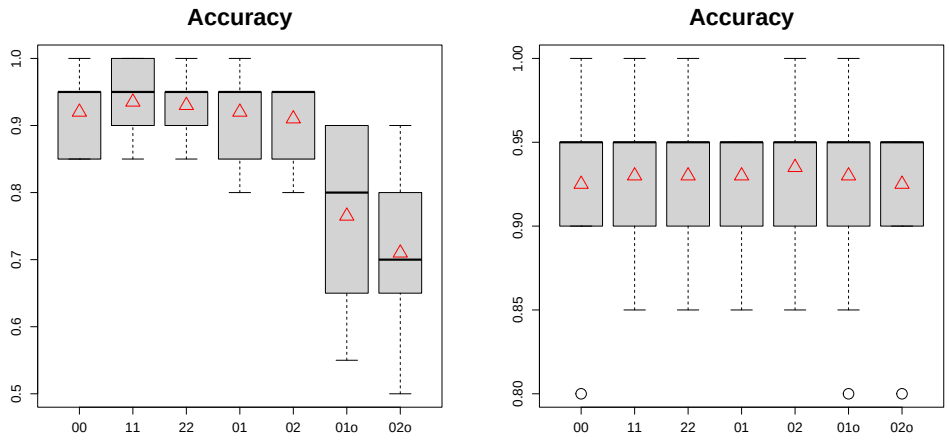

Figure 7: Box plots of Accuracy values of SF-MK-SVM given by a 10-fold cross-validation using the linear kernel (left) and the Gaussian kernel (right).

case, is that the weight optimization might suffer from over-fitting. We shall see in the next section that on some real datasets SF-MK-SVM using weight optimization can improve the performances.

4.2. Real-world data

In this paragraph, we report on the performances of SF-MK-KM and SF-MK-SVM on the 6 real-world datasets whose characteristics are given in Table 1. These data are all publicly available and come from either the fda R package [48], the fda.usc R package [49] or the UEA and UCR TS Classification Repository [50]. Here is a brief description of the datasets :

- *Growth* : it contains measurements of the heights of 39 boys and 54 girls from age 1 to 18. The measurements are taken at regular intervals. The task consists in separating boys and girls growth curves.

- *Trace* : it is a synthetic dataset designed to simulate instrumentation failures in a nuclear power plant. There are 4 different transient classes corresponding to distinct curve shapes.

- *poblenou* : it corresponds to NOx levels measured every hour by a control station in Poblenou in Barcelona (Spain). The goal is to dis-

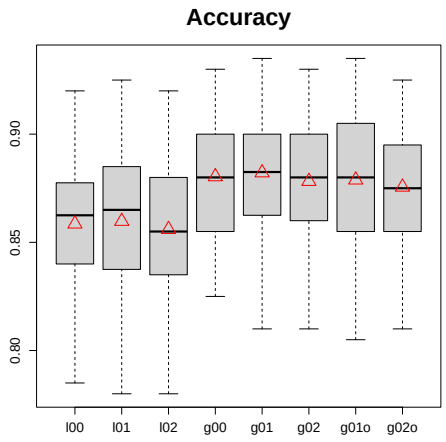

Figure 8: Box plots of averaged Accuracy values of SF-MK-SVM tested on 100 samples of 200 curves (100 in each group) using linear and Gaussian kernels. The prefix l and g respectively stand for linear and Gaussian kernels. The suffix o stands for weights optimization.

criminate air pollution trajectories during working days from the ones during non-working days.

- *Meat* : it concerns food spectrographs used in chemometrics to classify food types. The data are obtained using Fourier transform infrared (FTIR) spectroscopy with attenuated total reflectance (ATR) sampling. There are 3 classes : chicken, pork and turkey.

- *phoneme* : it contains 250 speech frames with class membership: "sh", "iy", "dcl", "aa" and "ao". From each speech frame, a log-periodogram of length 150 has been stored. The goal is to predict the class membership.

- *SwedishLeaf* : it is a set of swedish tree leaf outlines where contour images are transformed into time series. There are 15 different species. We used the 500 observations of the test subset provided in the dataset repository.

We explained in sub-section 2.1, the pre-processing procedure we apply in order to reconstruct the functional form of the discretized observations.

Source	Type	Name	Nb of FD	Nb of Class	Nb of time pts
fda	Growth curve	*Growth*	93	2	31
UCR_TS	Sensor	*Trace*	100	4	275
fda.usc	Air pollution	*poblenou*	115	2	24
UCR_TS	Spectroscopy	*Meat*	120	3	448
fda.usc	Acoustic	*phoneme*	250	5	150
UCR_TS	Image	*SwedishLeaf*	500	15	128

Table 1: List of real-world datasets used in our experiments.

For all datasets, we used a B-splines basis systems of order $\mathrm{d} = \mathrm{q} + 2$ with $\mathrm{m} = \mathrm{d} + \mathrm{p}$ basis functions where p is the number of time points that depends on the dataset. A spline smoothing is carried out with a roughness penalty $R(x_i) = \|D^4 x_i\|^2_{\mathbb{L}_2}$ since we aim to study up to the 2nd derivative function (see for example [9, Chapter 5]). The hyper-parameter λ in (3) was selected among the values $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ by means of the GCV criterion.

For each dataset, we examined both the unsupervised and the supervised tasks using SF-MK-KM and SF-MK-SVM respectively. The setting is the same as for the simulated data. However, we experimented only with the Gaussian kernel. In that respect, the σ value in (24) needs to be adapted in regard to the different datasets. To this end, we adopted a general strategy for auto-tuning this hyper-parameter which is inspired from [51]. In the latter paper, the authors propose a local scaling for each pair $(x_i, x_{i'})$ and found that replacing σ^2 with $\sigma_i \sigma_{i'}$ in (24), where σ_i is the distance value from x_i to its 7th nearest-neighbor, was a good strategy in practice. However, the resulting affinity matrix is not guaranteed to be positive semi-definite. Yet, this latter condition is a requirement for the SVM model. In order to circumvent this issue, we determined the distribution of the distances to the 7th nearest neighbors of all elements and set the global σ value to the median estimate.

Regarding the clustering task, for each dataset and each representation 00, 11, 22, 01, 02, 01o, 02o, we provide the box plots of the Purity and NMI values given by 10 different initializations of SF-MK-KM. In the case of the classification task, the box plots are related to the values provided by the 10-fold cross-validation procedure.

The results of the clustering and classification performances of the distinct datasets are given in Figures 9, 10, 11, 12, 13, and 14.

In the subsequent paragraphs we comment on the results obtained for the clustering and the classification problems respectively in a synthesized fashion.

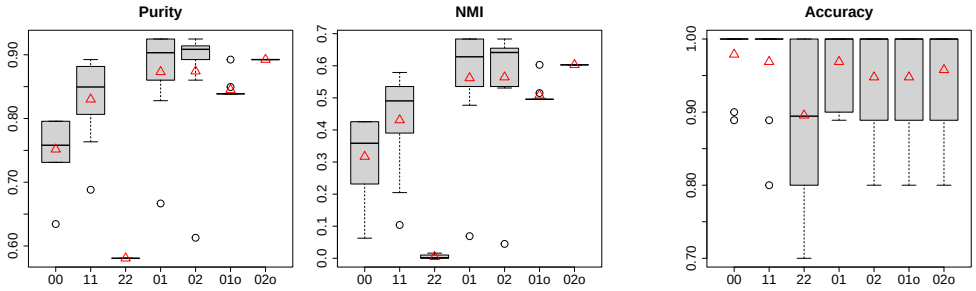

Figure 9: *Growth* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

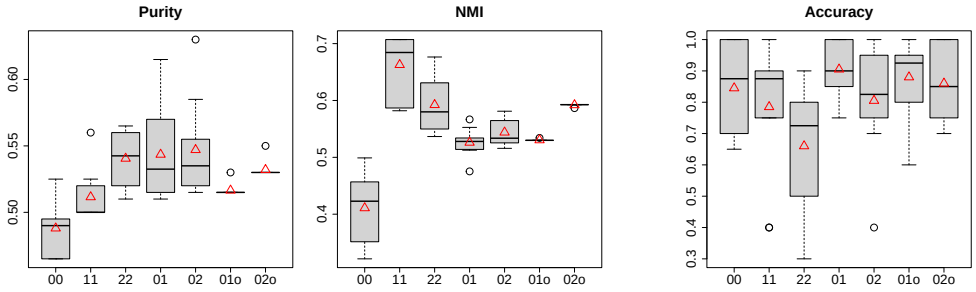

Figure 10: *Trace* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

4.2.1. Clustering tasks

The observations that we can draw from our experiments on real-world data for the unsupervised task are summarized below:

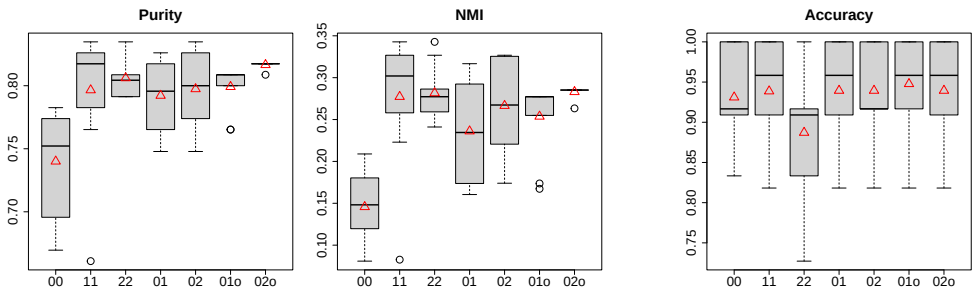

Figure 11: *poblenou* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

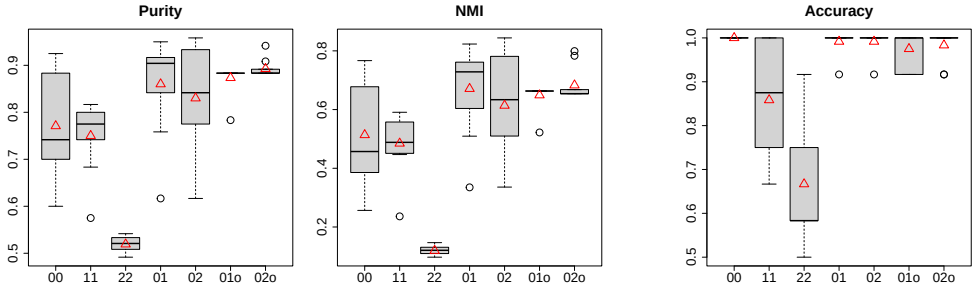

Figure 12: *Meat* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

- The individual scores of single kernel matrices 00, 11 and 22 can vary greatly. This situation exposes the practitioner to a certain risk when choosing only one among the three alternatives, for representing the FD.

- Combining the three different views allows one to reduce this risk. Indeed, the results reached by the representations 01 and 02 (no weight optimization) are close to the best scores among the cases 00, 11 and 22. This is the case for *poblenou*, *phoneme* and *SwedishLeaf*. For the remaining datasets, combining the different views allows improving the performances of the single representations. Indeed, for *Growth*, *Trace*

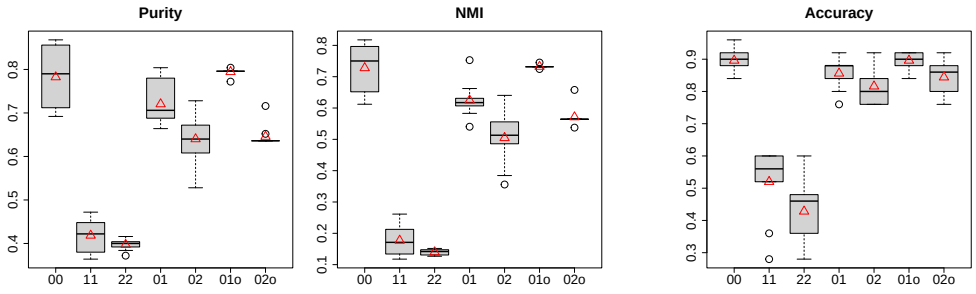

Figure 13: *phoneme* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

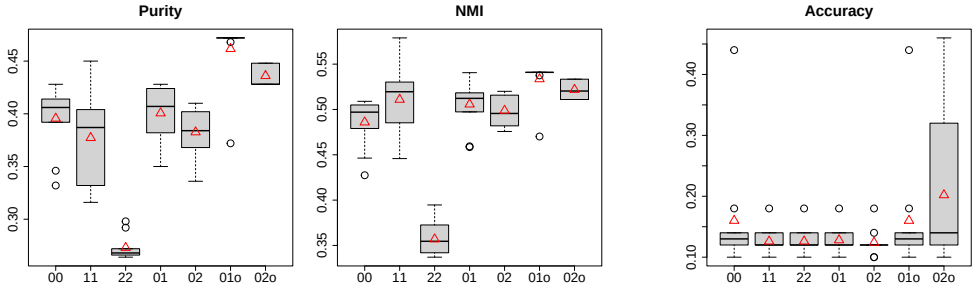

Figure 14: *SwedishLeaf* - Box plots of measures of Purity (left) and NMI (middle) given by SF-MK-KM using several initializations ; and box plot of measures of Accuracy (right) given by SF-MK-SVM using cross-validation.

(mean Purity values only) and *Meat*, the evaluation criteria of 01 and 02 tend to be higher than any measures obtained with 00, 11 or 22.

- In general, optimizing the weight and inferring a weighted Sobolev metric 01o and 02o enhances even more the values reached by 01 and 02 respectively. Exceptions concern *Growth* for which he Purity and NMI scores for 01o are lower than those of 01; and for *Trace* where the Purity values for 01o tend to be lower than for 01.

- For all datasets, the dispersion of the performances are tighter when weight optimization is performed. This was already underline in the

case of simulated data. It is an interesting feature of our framework : by maximizing the weights, the SF-MK-KM approach which relies on the k-means procedure is much less sensitive in regard to the partition random initialization.

- If we consider the mean or the median values of Purity or NMI, we can say that the setting 02o gives top performances for the following datasets : *Growth*, *poblenou*, *Meat*. For the cases, *phoneme* and *SwedishLeaf*, it is the 01o model that provides the best outcomes. Only for the *Trace* case, a single kernel matrix (11) outperforms all other representations. Yet, we note that the NMI score of 02o is the second best for this dataset.

- The question of fixing q to 1 or 2, is an open question. Without any expertise on this concern, we argue that choosing q = 2 and applying the weight optimization can be considered as a default robust strategy.

4.2.2. Classification tasks

As for as the supervised problem is concerned, we make the following comments on the results we obtained on the 6 datasets. In this case, our observations are based on the graph on the right of all Figures 9, 10, 11, 12, 13, and 14. We recall that the variability of the Accuracy is related to the 10-fold cross-validation.

- Similarly to the clustering task, it is less risky to use the multiple kernel matrices instead of choosing a single view, since the performances of 01 and 02 are, in many cases, close to the best among the ones reached by 00 or 11 or 22.

- However, unlike the unsupervised case, 01 and 02 rarely outperform the best performance among 00, 11 and 22. Only for *Trace* (01) and *poblenou* (01 and 02), SF-MK-SVM were able to improve the pattern recognition. Yet, as mentioned in the previous point, combining the distinct views provide assessment scores that are rather close to the best one observed among the single views.

- Weight optimization can improve the performances : the mean or median Accuracy scores obtained with 01o are greater than those of 01 for *Trace*, *poblenou*, *phoneme* and *SwedishLeaf*. When comparing 02 and

02o, we also note an increased performance for *Growth*, *Trace*, *poblenou*, *phoneme* and *SwedishLeaf*. In the latter case, the gain is particularly important. Indeed, 02o is the best model for the *SwedishLeaf* dataset. The only bad but limited impact of weight optimization is for *Growth* in the case of 01 *versus* 01o and for *Meat* in the context of 01 *versus* 01o as well.

- All tasks except the *SwedishLeaf* one, achieve very high accuracy rates. In the majority of cases, at least one representation among 00, 11 or 22 already allows a pretty precise pattern recognition. But, as emphasized in paragraph 4.2.1, without any expertise, one might not know which one of the three alternatives to choose. For the classification task too, we argue that a default robust strategy is to apply SF-MK-SVM with weight optimization.

- The *SwedishLeaf* dataset presents a rather high number of classes (15) as compared to the other cases. This feature might explain the difficulty of the learning task. Indeed, the scores reached by the single views are very low. Despite this fact, 02o was able to dramatically increase the Accuracy measure. Our supervised model that promotes weight optimization seems to be efficient in such cases but we need further experiments to confirm this point.

5. Conclusion and future work

FDA makes it possible to study continuous phenomena using statistical and machine learning approaches augmented with tools from functional analysis. In this paper, we assume that FD are elements of Sobolev spaces $\mathbb{W}^{q,2}$ and we apply successively the differential operator in order to obtain derivative functions up to order q. Then, we introduce learning methods that aim to infer a rich representation of FD in an appropriate functional space. To this end, we propose two main ingredients. Firstly, we apply kernel methods in order to implicitly map the FD and their derivative functions in separate RKHS. Secondly, we propose to learn how to combine the resulting kernel functions for unsupervised and supervised learning tasks. More precisely, our contribution from a methodological standpoint is twofold. On the one hand, we introduce SF-MK-KM (Sobolev Functions - Multiple Kernel - k-Means) for clustering problems. On the other hand, we extend to Sobolev functions the multiple kernel SVM method that we denote by SF-MK-SVM. Both

techniques amount to learn weighted Sobolev metrics where the derivative functions of order $s = 0, \ldots, \mathrm{q}$ can have different impacts.

In the goal of testing our methods, we experimented with simulated and real-world data. Our empirical work shows that applying a Gaussian kernel can improve the clustering and the classification performances in comparison to the linear kernel. Moreover, the weight optimization principle exhibit interesting features. For the clustering task, it refines even more the clustering outcomes. Furthermore, we empirically demonstrate that SF-MK-KM becomes less sensitive to the random initialization inherent to the underlying k-means procedure it is based upon. Regarding the classification problem, SF-MK-SVM with weight optimization does not always give the overall best accuracy scores but, in our experiments, its performances are generally close to the best representation among the single views $\{x_i\}_i$ xor $\{Dx_i\}_i$ xor $\{D^2x_i\}_i$. In the situation where a practitioner does not have any evidence for selecting one over the three alternatives, we argue that our methods, that learn how to combine the three views, are robust with respect to the unknown quality of the single representations for clustering or classification purposes.

As future work, we intend to extend our framework by using weight functions rather than weight scalars for balancing each derivation order. In that perspective, the sparse clustering framework designed in [5] and the interpretable SVM technique for FD introduced in [52] would be worth considering. Besides, it is worth mentioning that the techniques that we have introduced can be naturally extended to more general scenarios. In particular, they can address multivariate functional data which are *de facto* multiview learning problems. Another intriguing research line is physics informed machine learning. In this context, we are interested in examining a principled way to integrate more general differential operators in our framework in order to leverage physics constraints in the learning procedure.

Acknowledgment

This work was partly supported by the Agence Nationale de la Recherche of the French government through the program "Investissements d'Avenir" ANR-10-LABX-14-01.

References

[1] C. Abraham, P.-A. Cornillon, E. Matzner-Løber, N. Molinari, Unsupervised curve clustering using b-splines, Scandinavian journal of statistics 30 (3) (2003) 581–595.

[2] T. Tarpey, K. K. Kinateder, Clustering functional data, Journal of classification 20 (1) (2003) 093–114.

[3] J.-M. Chiou, P.-L. Li, Functional clustering and identifying substructures of longitudinal data, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69 (4) (2007) 679–699.

[4] M. L. L. García, R. García-Ródenas, A. G. Gómez, K-means algorithms for functional data, Neurocomputing 151 (2015) 231–245.

[5] D. Floriello, V. Vitelli, Sparse clustering of functional data, Journal of Multivariate Analysis 154 (2017) 1–18.

[6] F. Rossi, B. Conan-Guez, A. El Golli, Clustering functional data with the som algorithm., in: ESANN, 2004, pp. 305–312.

[7] F. Rossi, N. Delannay, B. Conan-Guez, M. Verleysen, Representation of functional data in neural networks, Neurocomputing 64 (2005) 183–210.

[8] F. Rossi, N. Villa, Support vector machine for functional data classification, Neurocomputing 69 (7-9) (2006) 730–742.

[9] J. Ramsay, B. Silverman, Functional Data Analysis, Springer Science & Business Media, 2005.

[10] J. Dauxois, A. Pousse, Y. Romain, Asymptotic theory for the principal component analysis of a vector random function: some applications to statistical inference, Journal of multivariate analysis 12 (1) (1982) 136–154.

[11] B. S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster analysis 5th ed (2011).

[12] J. Jacques, C. Preda, Functional data clustering: a survey, Advances in Data Analysis and Classification 8 (3) (2014) 231–255.

[13] D. B. Hitchcock, M. C. Greenwood, Clustering functional data, in: Handbook of Cluster Analysis, Chapman and Hall/CRC, 2015, pp. 286–309.

[14] G. Biau, L. Devroye, G. Lugosi, On the performance of clustering in hilbert spaces, IEEE Transactions on Information Theory 54 (2) (2008) 781–790.

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

[16] D. M. Witten, R. Tibshirani, A framework for feature selection in clustering, Journal of the American Statistical Association 105 (490) (2010) 713–726.

[17] F. Ferraty, P. Vieu, Nonparametric functional data analysis: theory and practice, Springer Science & Business Media, 2006.

[18] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, The Annals of Statistics (1995) 73–102.

[19] G. M. James, T. J. Hastie, Functional linear discriminant analysis for irregularly sampled curves, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (3) (2001) 533–550.

[20] F. Chamroukhi, H. D. Nguyen, Model-based clustering and classification of functional data, WIREs Data Mining and Knowledge Discovery 9 (4) (2019).

[21] G. M. James, Generalized linear models with functional predictors, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64 (3) (2002) 411–432.

[22] H.-G. Müller, U. Stadtmüller, et al., Generalized functional linear models, Annals of Statistics 33 (2) (2005) 774–805.

[23] G. Fan, J. Cao, J. Wang, Functional data classification for temporal gene expression data with kernel-induced random forests, in: 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE, 2010, pp. 1–5.

[24] A. Möller, G. Tutz, J. Gertheiss, Random forests for functional covariates, Journal of Chemometrics 30 (12) (2016) 715–725.

[25] F. Rossi, B. Conan-Guez, Theoretical properties of projection based multilayer perceptrons with functional inputs, Neural Processing Letters 23 (1) (2006) 55–70.

[26] T.-Y. Hsieh, Y. Sun, S. Wang, V. Honavar, Functional autoencoders for functional data representation learning, in: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), SIAM, 2021, pp. 666–674.

[27] N. Krämer, Boosting for functional data, arXiv preprint math/0605751 (2006).

[28] K. Fuchs, J. Gertheiss, G. Tutz, Nearest neighbor ensembles for functional data with interpretable feature selection, Chemometrics and Intelligent Laboratory Systems 146 (2015) 186–197.

[29] C. Preda, Regression models for functional data by reproducing kernel hilbert spaces methods, Journal of statistical planning and inference 137 (3) (2007) 829–840.

[30] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, J. Audiffren, Operator-valued kernels for learning from functional response data, Journal of Machine Learning Research 17 (20) (2016) 1–54.

[31] G. Biau, F. Bunea, M. H. Wegkamp, Functional classification in hilbert spaces, IEEE Transactions on Information Theory 51 (6) (2005) 2163–2172.

[32] P. Besse, J. O. Ramsay, Principal components analysis of sampled functions, Psychometrika 51 (2) (1986) 285–311.

[33] F. Ferraty, P. Vieu, Curves discrimination: a nonparametric functional approach, Computational Statistics & Data Analysis 44 (1-2) (2003) 161–173.

[34] F. Ieva, A. M. Paganoni, D. Pigoli, V. Vitelli, Multivariate functional clustering for the morphological analysis of electrocardiograph curves, Journal of the Royal Statistical Society: Series C (Applied Statistics) 62 (3) (2013) 401–418.

[35] Y. Meng, J. Liang, F. Cao, Y. He, A new distance with derivative information for functional k-means clustering algorithm, Information Sciences 463 (2018) 166–185.

[36] T. Villmann, Sobolev metrics for learning of functional data - mathematical and theoretical aspects, Machine Learning Reports, Research group on Computational Intelligence (2007).

[37] A. M. Alonso, D. Casado, J. Romo, Supervised classification for functional data: A weighted distance approach, Computational Statistics & Data Analysis 56 (7) (2012) 2334–2346.

[38] A. Ahmedou, J.-M. Marion, B. Pumo, Generalized linear model with functional predictors and their derivatives, Journal of Multivariate Analysis 146 (2016) 313–324.

[39] F. Rossi, N. Villa-Vialaneix, Consistency of functional learning methods based on derivatives, Pattern Recognition Letters 32 (8) (2011) 1197–1209.

[40] G. Kimeldorf, G. Wahba, Some results on tchebycheffian spline functions, Journal of mathematical analysis and applications 33 (1) (1971) 82–95.

[41] S. Bickel, T. Scheffer, Multi-view clustering, in: Fourth IEEE International Conference on Data Mining (ICDM'04), IEEE, 2004, pp. 19–26.

[42] Y. Yang, H. Wang, Multi-view clustering: A survey, Big Data Mining and Analytics 1 (2) (2018) 83–107.

[43] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, Y. Moreau, Optimized data fusion for kernel k-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (5) (2011) 1031–1039.

[44] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: 2012 IEEE 12th international conference on data mining, IEEE, 2012, pp. 675–684.

[45] J. C. Bezdek, Fuzzy Mathematics In Pattern Classification., Cornell University, 1973.

[46] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, A. Zien, Efficient and accurate lp-norm multiple kernel learning., in: NIPS, Vol. 22, 2009, pp. 997–1005.

[47] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, Non-sparse regularization and efficient training with multiple kernels (2010).

[48] J. O. Ramsay, H. Wickham, S. Graves, G. Hooker, fda: Functional data analysis, R package version 2 (4) (2014) 142.

[49] M. Febrero-Bande, M. Oviedo de la Fuente, Statistical computing in functional data analysis: The R package fda.usc, Journal of Statistical Software 51 (4) (2012) 1–28.
URL `https://www.jstatsoft.org/v51/i04/`

[50] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Mining and Knowledge Discovery 31 (2017) 606–660.

[51] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in neural information processing systems, 2005, pp. 1601–1608.

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