Highlights

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

Julien Ah-Pine, Anne-Françoise Yao

- Representing functions and derivatives in RKHS by means of kernel methods can improve pattern recognition in both unsupervised and supervised learning tasks.
- Learning how to combine kernel functions that represent functions and their derivatives in a complementary manner can boost clustering and classification performances.
- Our MK-KM-FD and MK-SVM-FD are respectively extensions of the multiple kernel k-means and multiple kernel SVM from vectors to functions with derivatives.

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

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

[a] *Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, 63000 Clermont–Ferrand, France.*
[b] *Université Clermont Auvergne, CNRS, LMBP, 63000 Clermont–Ferrand, France.*
[c] *Université Clermont Auvergne, CNRS, IRD, CERDI, 63000 Clermont–Ferrand, France.*

Abstract

In this paper, we propose a framework for rich representation of smooth functional data, leveraging a multiview approach that considers functions and their derivatives as complementary sources of information. Additionally, motivated by the non-linear nature of functional data, we advocate for kernel methods as a suitable modeling approach. We extend existing multiple kernel learning techniques for multivariate data to handle functional data. In particular, we introduce a general procedure for linearly combining different kernel functions. We apply this framework to both clustering and classification tasks, extending multiple kernel k-means and multiple kernel SVM methods to Sobolev functions in $\mathbb{H}^q$. Our experiments involve both simulated and real-world data, demonstrating the effectiveness of our proposed methods.

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

1. Introduction

Modern technologies allow for the massive recording of 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, among others. These measurements are associated with timestamps and/or geographical locations and are recorded as discrete data. However, they actually represent discretized observations of continuous curves or surfaces. From a data analysis standpoint, it can be advantageous to consider the continuous nature of the phenomenon under study rather than solely analyzing the discrete observations. In particular, working with continuous functions allows for the leveraging of tools from functional analysis such as differential operators. Functional Data Analysis (FDA) is the branch of statistics and data science 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) both for unsupervised and supervised tasks. In this paper, we propose to investigate the multiple kernel paradigm for clustering and classifying FD with derivatives. Our motivations are as follows. Firstly, we aim to exploit the possibility of utilizing derivative functions in order to obtain a richer representation of FD. In that perspective, we assume that the FD belong to the Sobolev space $\mathbb{H}^{\mathrm{q}}([0,T])$. In that case, the successive derivatives up to order q exist in the weak sense and the latter provide as many distinct sources of information that one can leverage. Indeed, one can combine these different views to obtain a richer geometric representation of the FD for pattern recognition purposes. Secondly, similar to multivariate data, we argue that projecting FD and derivatives onto Reproducing Kernel Hilbert Spaces (RKHS) can be beneficial in the non-linear case. In that context, we advocate for multiple kernel learning techniques. Functional data are handled by extending the multiple kernel k-means method for clustering problems, and the multiple kernel support vector machine (SVM) technique in classification.

The rest of the paper is organized as follows. In section 2, we recall background materials on FDA and introduce our general framework for representing functions with derivatives. In addition to using kernels, we also present a general optimization procedure for learning how to balance the information conveyed by the different derivatives. Then, in section 3, we address the FD clustering task. After providing a brief overview of previous works, we detail our extension of the multiple kernel k-means to deal with FD with derivatives. This section also exposes the experiments we conducted on artificial and real-world data. Section 4 focuses on the classification task. It follows the same structure as section 3. In the supervised case, we propose an extension of the multiple kernel SVM technique for FD with derivatives. Finally, we sum up the main points of our contributions and briefly discuss future works in section 5.

2. Multiple kernel representation of functions with derivatives

We review past research works in FDA that utilize derivatives as alternative or complementary views of FD. Then, we present our global framework that advocates for the application of kernel methods. Our approach also includes a general procedure for inferring the optimal weight distribution when linearly combining the distinct views provided by the derivative functions.

2.1. Previous works on functional data analysis with derivatives

When clustering and classifying FD, one can exploit the derivative functions, which encode additional discriminant information such as slopes or curvatures. This was pointed out at least since [1] in the FDA community. From a conceptual standpoint, semi-metrics derived from derivative functions were particularly emphasized in [2, 3]. In the data mining community the use of derivatives in place of the original curves was proposed first by [4]. We also mention [5] where the authors utilized a combination of distances between curves and distances between derivatives for time series classification.

The usefulness of derivatives for FD pattern recognition has been empirically demonstrated in several research works. Regarding FD clustering, [6, 3] show that for spectrometric data, the 2nd derivative functions can be more appropriate than the original functions. In the case of electrocardiograph curves, [7] demonstrates that a composite distance measure, which simply adds distances between the original curves and distances between the 1st derivatives, improves the performance of the k-means algorithm. Similarly, in [8], it is shown that the k-means algorithm can perform better with a composite distance using up to the 2nd derivatives. Both aforementioned papers apply uniform weights when aggregating the distance measures. In contrast, the framework introduced in [9] highlights the use of non-uniform weights. However, the question of estimating the weights remains open.

In the context of classification problems, several research works have promoted the use of semi-metrics as well. In the context of binary classification, the authors of [10] propose a framework based on Linear Discriminant Analysis (LDA). Other multivariate statistical methods extended to FD have also been examined with the addition of derivative functions. For example, in [11], the functional logistic regression and, more generally, the generalized functional linear model is examined with derivative functions included as functional covariates. Concerning machine learning techniques, we also mention the nearest neighbors based approach introduced in [12]. Additionally, we cite [13], which studies, from a theoretical standpoint, supervised learning tasks involving FD in the Sobolev space $\mathbb{H}^{\mathrm{q}}([0,T])$.

To the best of our knowledge, none of the previous work in FDA has proposed a functional representation framework that combines a multiview perspective using successive derivatives, implicit non-linear projections of curves and derivatives, and non-uniform weighting schemes of kernel functions altogether.

2.2. Representing functions with derivatives using kernels

In this contribution, we assume that the objects under study are 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 $T > 0$. Here, $\mathbb{H}^{\mathrm{q}}([0,T])$ (also denoted as $\mathbb{H}^{\mathrm{q}}$ subsequently) represents the Sobolev space of functions whose derivatives, in the weak sense, up to order q are elements of the Hilbert space $\mathbb{L}^2([0,T])$ (also denoted as $\mathbb{L}^2$ subsequently).

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

where D is the differential operator.

Given the sample $\{x_i\}_{i=1,\dots,\mathrm{n}}$, the sets of derivative functions up to order q are respectively denoted $\{D^1 x_i\}_i$, $\{D^2 x_i\}_i$, $\dots$, $\{D^{\mathrm{q}} x_i\}_i$. These sets of functions are interpreted as distinct views of the same objects. It is important to note that, although we suppose that the FD are elements of $\mathbb{H}^{\mathrm{q}}$, we do not restrict ourselves to the regular Sobolev metric:

$$\langle x_i, x_{i'} \rangle_{\mathbb{H}^{\mathrm{q}}} = \sum_{s=0}^{\mathrm{q}} \langle D^s x_i, D^s x_{i'} \rangle_{\mathbb{L}^2}, \quad \forall i, i' = 1, \dots, \mathrm{n},$$

where D^0 is the identity operator.

We consider a metric for each order $s = 0, 1, \dots, \mathrm{q}$, by employing (possibly) distinct kernel functions $k^s : \mathbb{L}^2 \times \mathbb{L}^2 \to \mathbb{R}$, in place of the usual $\mathbb{L}^2$ inner product. Therefore, for any pair $(x_i, x_{i'})$, we promote the use of:

$$k^s(D^s x_i, D^s x_{i'}), \quad \forall s = 0, \dots, \mathrm{q}.$$

In this case, it should be noted that the usual notion of Sobolev space is no longer valid. Indeed, the representations of the derivatives $\{D^s x_i\}_i$ with $s > 0$ in their respective feature spaces associated to k^s do not generally correspond to the derivatives of the representation of the functions $\{x_i\}_i$ in their feature space associated to k^0.

Furthermore, we advocate for non-uniform weights, supposing that some sets of derivatives are more discriminant than other ones and should be more emphasized. Consequently, given any pair of functions $(x_i, x_{i'})$, we adopt the more general metric:

$$k(x_i, x_{i'}) = \sum_{s=0}^{\mathrm{q}} w_s k^s(D^s x_i, D^s x_{i'}), \quad \text{where } w_s \geq 0, \forall s = 0, \ldots, \mathrm{q}. \tag{2}$$

For all $s = 0, \ldots, \mathrm{q}$, w_s is the non-negative weight assigned to the information conveyed by $\{D^s x_i\}_i$, which consists of derivatives of order s.

2.3. Learning how to linearly combine kernel matrices of derivative functions

2.3.1. $\mathbf{w}$ *as the analytical solution of an optimization problem*

We introduce a simple optimization problem that is central to our framework. The material is presented from a general scope. We detail latter on, in sections 3 and 4, how it specifically applies to our clustering and classification models, respectively.

Suppose that we are given a non-negative vector $\mathbf{z} = (z_s)_{s=1,\ldots,\mathrm{q}}$, where z_s is seen as the partial profit of view s. We aim to maximize the overall profit by means of a linear combination $\sum_{s=0}^{\mathrm{q}} w_s z_s$, under the constraints $w_s \geq 0, \forall s = 0, \ldots, \mathrm{q}$, and $\|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1$. The latter condition is aimed to bound the problem.

In our FDA context, this problem represents, from a general perspective, how the information provided by the first q derivatives is integrated. It is carried out by maximizing $\sum_{s=0}^{\mathrm{q}} w_s z_s$, where z_s is the (non-negative) gain associated to using derivative s.

Thereby, we seek to solve the following optimization problem:

$$\max_{\mathbf{w} \in \mathbb{R}^{\mathrm{q}+1}} \mathbf{w}^\top \mathbf{z} \quad \text{s.t.} \quad \begin{cases} \mathbf{w} \geq \mathbf{0}, \\ \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1. \end{cases} \tag{3}$$

where $\mathbf{w} \geq \mathbf{0}$ is a shortcut for $w_s \geq 0, \forall s = 0, \ldots, \mathrm{q}$.

The norm hyper-parameter r can be chosen in the interval $[1, \infty]$. However, we discard the case $\mathrm{r} = 1$ because it would assign all the weight to the view with the maximum partial profit. Our hypothesis is that the derivative functions of various orders are complementary to each other and our objective is to design an aggregation scheme rather than a selection strategy.

Problem (3) is convex and its closed-form solution is given below.

Proposition 1. *Assuming* $\mathbf{z} \geq \mathbf{0}$ *and* $\mathrm{r} > 1$*, the solution to Problem* (3) *is given by:*

$$w_s^* = \frac{z_s^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=0}^{\mathrm{q}} z_{s'}^{\frac{\mathrm{r}}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}}}, \quad \forall s = 0, \ldots, \mathrm{q}. \tag{4}$$

The proof is provided in appendix. However, it is worth mentioning that similar optimization problems have been studied in the literature for the multivariate case. We discuss these related works in the following sub-section to highlight the difference with our context.

2.3.2. Related works in multiple kernel learning in the multivariate case

In [14], the authors introduced a multiple kernel k-means algorithm for multivariate data. In this work as well, an optimization problem is analytically solved in order to learn how to linearly combine the different kernel matrices. More precisely, given a vector $\mathbf{d} = (d_s)_{s=0,\ldots,\mathrm{q}}$ of non-negative cost values, the views' weights vector $\mathbf{v} = (v_s)_{s=0,\ldots,\mathrm{q}}$ is determined by solving:

$$\min_{\mathbf{v} \in \mathbb{R}^{\mathrm{q}+1}} \sum_{s=0}^{\mathrm{q}} v_s^{\mathrm{r}} d_s \quad \text{s.t.} \quad \begin{cases} \mathbf{v} \geq \mathbf{0}, \\ \sum_{s=0}^{\mathrm{q}} v_s = 1. \end{cases} \tag{5}$$

For $\mathrm{r} > 1$, the closed-form solution of Problem (5) is given by[1] [15, 14]:

$$v_s^* = \frac{1}{\sum_{s'=0}^{\mathrm{q}} \left(\frac{d_s}{d_{s'}}\right)^{\frac{1}{\mathrm{r}-1}}} = \frac{\left(\frac{1}{d_s}\right)^{\frac{1}{\mathrm{r}-1}}}{\sum_{s'=0}^{\mathrm{q}} \left(\frac{1}{d_{s'}}\right)^{\frac{1}{\mathrm{r}-1}}}, \quad \forall s = 0, \ldots, \mathrm{q}. \tag{6}$$

Problem (5) and its solution (6) appear similar to Problem (3) and its solution (4). In fact, if we assume that the quantities used in Problem (3) and Problem (5) are related as follows, $z_s = (1/d_s)^{1/\mathrm{r}}, \forall s = 0, \ldots, \mathrm{q}$, then w_s^* is exactly $(v_s^*)^{1/\mathrm{r}}$. However, in [14], d_s represents the within cluster variance associated with view s, whereas in our perspective, as we shall explain in section 3, z_s represents the between cluster variance conveyed by view s.

[1] It is noteworthy that Problem (5) and its solution (6) were originally introduced by Bezdek in [15, 16]. However, the problem was designed to determine the membership values of objects to clusters in the context of the fuzzy c-means procedure.

Between and within cluster variance measures do vary in opposite direction but in a linear fashion. They do not satisfy the relationship $z_s = (1/d_s)^{1/\mathrm{r}}$, therefore Problems (3) and (5) provide different solutions.

In the case of multiple kernel learning with SVM for multivariate data, our approach represents a particular instance of the framework detailed in the works [17, 18]. As we shall see in section 4, z_s is a specific evaluation of the quadratic form associated to the kernel matrix of view s. In this case, (4) becomes (27) which is the same formula given in [17, Corollary 3].

Despite the existence of previous related contributions, to our knowledge, the application of Proposition 1 for aggregating the information from functions and their successive derivatives in the context of FDA and multiple kernel machines is new.

2.4. Reconstructing the functional data from discrete observations

Before delving into our clustering and classification models, let us first revisit some basic pre-processings in FDA. In practice, one typically does not directly observe entire curves but rather samples of their realizations at different time points in the interval $[0, T]$. Therefore, when analysing real-world data, it is necessary to reconstruct an approximate functional form using the finite and discrete set of values.

While the set of observation points for two distinct FD, x_i and $x_{i'}$, can be different, we suppose that all FD were measured with respect to the same time grid $\{t_j\}_{j=1,\dots,\mathrm{p}}$. Consequently, for all x_i, $i = 1, \dots, \mathrm{n}$, we have p observations $\{y_{ij}\}_{j=1,\dots,\mathrm{p}}$. However, we presume that these measurements could have been corrupted by noise. Hence, we suppose that:

$$y_{ij} = x_i(t_j) + \epsilon_{ij}, \quad \forall i = 1, \dots, \mathrm{n}, \forall j = 1, \dots, \mathrm{p}, \tag{7}$$

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

To infer approximated functional forms for $\{x_i\}_i$ departing from $\{y_{ij}\}_{i,j}$, we suppose that the FD can be represented as linear combinations of a predefined set of basis of functions. In this context, we consider the commonly used B-splines basis system which consists of polynomial functions. Since we assume that the derivatives up to the q^{th} order are in $\mathbb{L}^2$, we work with the subspace of functions spanned by the set of B-splines of order $\mathrm{d} = \mathrm{q} + 2$ to ensure a sufficiently rich framework to represent the functional data and their derivatives. Consequently, the basis system has a dimension of $\mathrm{m} = \mathrm{d} + \mathrm{p}$.

Let $\{\phi_k\}_{k=1,\dots,\mathrm{m}}$ be a set of m B-splines that we denote in vector form as $\boldsymbol{\phi} = (\phi_k)_{k=1,\dots,\mathrm{m}}$. Therefore, the approximated FD are assumed to be elements of the subspace $\mathrm{Span}(\phi_1, \dots, \phi_\mathrm{m}) \subset \mathbb{H}^\mathrm{q}$:

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

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

It is important to underline that using a set of smooth basis functions $\{\phi_k\}_k$ facilitates the determination of the successive derivative functions of $\{x_i\}_i$. Since the differential operator D is linear, it is sufficient to determine the sets of derivatives of the basis functions $\{D^s\phi_k\}_k$ for $s = 1, \dots, \mathrm{q}$.

For each element x_i, one must estimate $\mathbf{c}_i$ based on the observations $\{y_{ij}\}_{j=1,\dots,\mathrm{p}}$. Due to noise corruption, this problem is typically addressed using least squares. Additionally, to prevent overfitting and achieve better control over the smoothness of the FD, a roughness penalty term denoted R is applied. Assuming the need for up to the q$^{\mathrm{th}}$ derivatives, one can set $R(x_i) = \|D^{\mathrm{q}+2}x_i\|^2_{\mathbb{L}_2}$ as suggested in [19, Chapter 5].

More formally, the spline smoothing procedure that estimates $\mathbf{c}_i$ for the FD x_i involves solving:

$$\hat{\mathbf{c}}_i = \underset{\mathbf{c}\in\mathbb{R}^\mathrm{m}}{\arg\min} \sum_{j=1}^{\mathrm{p}} (y_{ij} - x_i(t_j))^2 + \lambda R(x_i), \tag{8}$$

where $x_i(t_j) = \sum_{k=1}^{\mathrm{m}} c_{i,k}\phi_k(t_j)$ and $\lambda > 0$ is a tuning hyper-parameter estimated by a cross-validation procedure. In this paper, we employ the generalized cross-validation (GCV) criterion for this purpose.

3. Multiple kernel clustering of functions with derivatives

We begin by providing a brief overview of previous research activities in functional data clustering. Subsequently, we introduce our approach which focuses on FD with derivatives and where each set of derivatives is considered a distinct view of the same objects. Utilizing kernel functions enables us to address non-linearity. Specifically, we extend the multiple kernel k-means technique to functions with derivatives. In that perspective, we explain how Proposition 1, as discussed in sub-section 2.3, is applied to automatically update the views' weights during the partitioning process. To demonstrate the efficacy of our clustering framework, we present experimental results from both simulated and real-world datasets.

3.1. Previous works on functional data clustering

Data clustering aims to automatically group a set of n items into subsets called clusters, forming a partition. The goal is to ensure that members within the same cluster are more similar to each other than to elements in other clusters. Assume that we have k clusters then the partition is denoted $C = \{C_1, \ldots, C_\mathrm{k}\}$. Many multivariate clustering methods have been adapted or extended in order to address FD. Reviews of these approaches can be found in [20, 21].

In this contribution, we focus on the functional k-means algorithm. Pioneering works in this area include [22] and [23]. In the former, FD are projected onto a set of B-splines similar to (8) while the latter focuses on Gaussian random functions. In [24], a re-assignment procedure similar to k-means was conducted, where each x_i is compared to its projection on the truncated Karhunen-Loève expansion of each cluster. Theoretical analysis of the k-means problem in Hilbert spaces is presented in [25]. Additionally, in [26], the k-means algorithm is carried out with the FD being represented utilizing a set of basis functions of an RKHS with a kernel function defined on $[0, T] \times [0, T]$. Another related paper is [27], where the k-means partitions the curves while a weight function defined on $[0, T]$ is learned to select sub-intervals that favor the variance.

All those research works apply the basic steps of the k-means algorithm, with the main differences lying in the representation used for FD. Our contribution diverges from these previous approaches by integrating the derivative functions in the FD representation. Furthermore, we operate under the assumption that FD and their derivatives may belong to non-linear subspaces. This is related to the manifold hypothesis: even if data are described in high dimensional linear spaces, in practice, it is often the case that they belong to non-linear manifolds with lower dimensions. Similar to multivariate data, we hypothesize that employing kernel functions[2] to implicitly project the FD on another space can be advantageous. In addition, introducing non-uniform weights to optimally blend the information coming from the derivative functions of different orders represents an innovative aspect of our work.

[2] Note that unlike the aforementioned paper [26], we use kernel functions defined on $\mathbb{L}_2([0, T]) \times \mathbb{L}_2([0, T])$ and not on $[0, T] \times [0, T]$.

3.2. The multiple kernel k*-means for FD with derivatives*

We propose to extend the multiple kernel k-means algorithm from the multivariate case to functions in $\mathbb{H}^{\mathrm{q}}$, where each of the derivatives of order $s = 0, \dots, \mathrm{q}$ is considered as a distinct view. More formally, the optimization problem that we are interested in is:

$$\min_{C,\mathbf{w}} \frac{1}{\mathrm{n}} \sum_{l=1}^{\mathrm{k}} \frac{1}{2|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s} \tag{9}$$
$$\text{s.t.} \begin{cases} C = \{C_1, \dots, C_{\mathrm{k}}\} \text{ is a partition,} \\ \mathbf{w} \geq \mathbf{0}, \\ \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1, \end{cases}$$

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

The loss function is the within cluster variance, which is a weighted mean of the within variance of each cluster C_l with $l = 1, \dots, \mathrm{k}$. In order to establish a graph-based formulation relying on kernel functions, we express the within variance in a pairwise manner. Moreover, we explicitly state that the objective function is separable with respect to the different views $s = 0, \dots, \mathrm{q}$.

By decomposing the total variance into the sum of the within and between cluster variances, we can alternatively maximize the between cluster variance. As a consequence, the previous problem is equivalent to the following one:

$$\max_{C,\mathbf{w}} \frac{1}{2\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s} \tag{10}$$
$$- \frac{1}{\mathrm{n}} \sum_{l=1}^{\mathrm{k}} \frac{1}{2|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \sum_{s=0}^{\mathrm{q}} w_s \|\psi^s(D^s x_i) - \psi^s(D^s x_{i'})\|^2_{\mathbb{F}^s}$$
$$\text{s.t.} \begin{cases} C = \{C_1, \dots, C_{\mathrm{k}}\} \text{ is a partition,} \\ \mathbf{w} \geq \mathbf{0}, \\ \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1. \end{cases}$$

Next, for all pairs $\{(x_i, x_{i'})\}_{i,i'=1,\dots,\mathrm{n}}$ and all views $s = 0, \dots, \mathrm{q}$, let us denote $\langle \psi^s(D^s x), \psi^s(D^s x') \rangle_{\mathbb{F}^s}$ by $k^s(D^s x_i, D^s x_{i'})$, and gather all these values in the kernel matrices $\mathbf{K}^s = (\mathbf{K}^s_{ii'})_{i,i'=1,\dots,\mathrm{n}} = (k^s(D^s x_i, D^s x_{i'}))_{i,i'}$. Then, if we expand the squared distances in Problem (10), it is not difficult to show that we obtain the following equivalent formulation:

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

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

We employ the standard strategy for solving such kinds of multiple kernel learning problems, which involves alternating between (i) maximizing with respect to C while keeping $\mathbf{w}$ fixed and (ii) maximizing with respect to $\mathbf{w}$ while keeping C fixed. In the former case, a usual kernel k-means algorithm is utilized to determine C. In the latter case, it is possible to reach a closed-form solution following the materials we exposed in sub-section 2.3. In that perspective, we introduce the following quantities:

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

Note that $z_s \geq 0, \forall s = 0, \ldots, \mathrm{q}$, since it corresponds to between cluster variance measures. Then, owing to Proposition 1, we have the following result.

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

$$\max_{\mathbf{w}} \sum_{s=0}^{\mathrm{q}} w_s \left(\sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'} \right) \tag{13}$$

$$\textit{s.t.} \begin{cases} \mathbf{w} \geq \mathbf{0}, \\ \|\mathbf{w}\|_{\ell_{\mathrm{r}}} \leq 1, \end{cases}$$

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

$$w^*_s = \frac{\left(\sum_{l=1}^{\mathrm{k}} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^s_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^s_{ii'} \right)^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=0}^{\mathrm{q}} \left(\sum_{l=1}^{k} \frac{1}{\mathrm{n}|C_l|} \sum_{i:x_i \in C_l} \sum_{i':x_{i'} \in C_l} \mathbf{K}^{s'}_{ii'} - \frac{1}{\mathrm{n}^2} \sum_{i=1}^{\mathrm{n}} \sum_{i'=1}^{\mathrm{n}} \mathbf{K}^{s'}_{ii'} \right)^{\frac{\mathrm{r}}{\mathrm{r}-1}} \right)^{\frac{1}{\mathrm{r}}}}. \tag{14}$$

It is worth mentioning that the range of each kernel values k^s for $s = 0, \ldots, \mathrm{q}$, can strongly vary. Accordingly, before combining the different kernel matrices $\{\mathbf{K}^s\}_{s=1,\ldots,\mathrm{q}}$, it may be important to carry out a normalization procedure to make them more comparable to each other.

We denote our method as MK-KM-FD, representing multiple kernel k-means for functions with derivatives. The procedure is summarized in Algorithm 1.

Algorithm 1: Multiple kernel k-means for functions with derivatives (MK-KM-FD).

```
Input: $\{y_{ij}\}_{i=1,\ldots,n;j=1,\ldots,\mathrm{p}}$ (sampled values of FD), $\mathrm{q} \geq 0$ (maximum
       order of derivative), $\mathrm{r} > 1$ ($\ell_\mathrm{r}$ norm, default 2), $\{k^s\}_{s=0,\ldots,\mathrm{q}}$
       (kernel functions, default Gaussian), $\sigma$ (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 (8);
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
```

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

3.3. Experiments with the MK-KM-FD model for functions with derivatives

3.3.1. Experiments settings

The research questions we investigate regarding the FD clustering task are as follows:

- Is it beneficial to combine functions with their derivatives by using a multiview approach ?
- Can we improve FD pattern recognition by projecting functions and derivatives in RKHS using non-linear kernel functions ?
- Does weights optimization allow for any improvement ?

We address these questions using the MK-KM-FD method described in Algorithm 1. We examined the cases q = 1 and q = 2, meaning that we take into account up to the 2nd derivatives. Regarding the constraint on $\mathbf{w}$'s ℓ_{r} norm, we set r = 2 in all our tests. The stopping condition in Algorithm 1 is triggered if a precision of 10^{-5} is reached for the objective function, or if a maximum of 10 iterations is achieved.

In order to study the different sets of derivatives either as single views or in a multiview framework, we tested MK-KM-FD using the following FD representations which rely only on *l*inear kernel functions:

- *l*0: $\mathbf{K}^{l0} = (\langle x_i, x_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\dots,\mathrm{n}}$,
- *l*1: $\mathbf{K}^{l1} = (\langle Dx_i, Dx_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\dots,\mathrm{n}}$,
- *l*2: $\mathbf{K}^{l2} = (\langle D^2x_i, D^2x_{i'} \rangle_{\mathbb{L}_2})_{i,i'=1,\dots,\mathrm{n}}$,
- *l*01: $\mathbf{K}^{l01} = \mathbf{K}^{l0} + \mathbf{K}^{l1}$,
- *l*012: $\mathbf{K}^{l012} = \mathbf{K}^{l0} + \mathbf{K}^{l1} + \mathbf{K}^{l2}$.

Note that $\mathbf{K}^{l01}$ and $\mathbf{K}^{l012}$ are equivalent to the regular Sobolev metric in $\mathbb{H}^1$ and $\mathbb{H}^2$, given in (2.2). For all previously exposed linear kernel matrices there is no weights optimization. This amounts to applying MK-KM-FD procedure without carrying out step 8 in Algorithm 1.

In contrast, the following kernel matrices result from a non-uniform linear combination of the distinct views:

- *l*01*o*: $\mathbf{K}^{l01o} = w_0\mathbf{K}^{l0} + w_1\mathbf{K}^{l1}$,
- *l*012*o*: $\mathbf{K}^{l012o} = w_0\mathbf{K}^{l0} + w_1\mathbf{K}^{l1} + w_2\mathbf{K}^{l2}$.

The weights are *o*ptimized at each iteration of Algorithm 1 using Corollary 1. In these cases, the full potential of our framework is exploited.

The acronyms from $l0$ to $l012o$, assigned to the previous kernel matrices, indicate the type of base kernel function (where l stands for $l$inear), the sets of derivatives involved in the combination (the digits denote the order of derivation taken into account) and whether the weights are updated at each iteration of not (where o stands for $o$ptimized weights).

To investigate the manifold hypothesis, we also utilized FD representations based on the Gaussian kernel in place of the linear kernel in all aforementioned cases. Therefore, for all possible single views $s = 0, 1, 2$ and all pairs of curves $(D^s x_i, D^s x_{i'})$ in the sample, we considered:

$$\mathbf{K}^{gs}_{ii'} = \exp\left(-\frac{\|D^s x_i - D^s x_{i'}\|^2_{\mathbb{L}_2}}{(\sigma^s)^2}\right), \quad \forall s = 0, 1, 2, \tag{15}$$

where $\sigma^s > 0$, is the hyper-parameter controlling the neighborhood width.

Consequently, the other batch of kernel matrices we experimented with is composed of: $\mathbf{K}^{g0}$, $\mathbf{K}^{g1}$, $\mathbf{K}^{g2}$ as single views; $\mathbf{K}^{g01}$, $\mathbf{K}^{g012}$, as multiple views; and $\mathbf{K}^{g01o}$, $\mathbf{K}^{g012o}$ as multiple views with weights optimization.

The clustering performances are assessed from an external validation perspective where we compare the partition C obtained by our clustering method against the ground-truth partition denoted by L. We use the normalized mutual information[3] (NMI) measure to assess the effectiveness associated with different representations of functions with derivatives. Let $L = \{L_1, \dots, L_\mathrm{k}\}$ and $C = \{C_1, \dots, C_\mathrm{k}\}$ denote the true classes and the found clusters respectively. Then, the NMI assessment measure is defined by:

$$\mathrm{NMI}(C, L) = \frac{2\mathrm{MI}(C, L)}{\mathrm{H}(C) + \mathrm{L}(C)} \tag{16}$$

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 $\mathrm{MI}(C, L) = \sum_{l,m=1}^{\mathrm{k}} (|C_l \cap L_m|/\mathrm{n}) \log(\mathrm{n}|C_l \cap L_m|/(|C_l||L_m|))$.

The NMI scores are in $[0, 1]$, where a higher value indicates a closer alignment between the two partitions and a better clustering solution.

MK-KM-FD relies on the kernel k-means heuristic. As this algorithm's random initialization often leads to different local optima, we ran MK-KM-FD multiple times with varying initializations to address this variability. To

[3] Note that we also employed the Purity measure as another evaluation criterion. We generally obtained similar conclusions as with NMI. As a result, we only expose the NMI measures in order to lighten the presentation of the experimental results.

ensure a robust assessment of differences between results obtained from two different FD representations, we applied paired t-tests with a significance level of 5% to compare the mean values of the obtained NMI measures.

3.3.2. Experiments with simulated data

In our first batch of experiments, we simulated univariate Gaussian density functions on the domain $[-4, 4]$. We considered two ground-truth clusters named group 1 and group 2, which are associated to two distinct sets of parameters $(\mu_1, \sigma_1) = (0, 1)$ and $(\mu_2, \sigma_2) = (0, 2)$ respectively. Furthermore, we introduced random noises for each parameter: $\epsilon_{\mu_1}, \epsilon_{\mu_2} \sim \mathcal{N}(0, 0.15)$, $\epsilon_{\sigma_1} \sim \mathcal{N}(1, 0.1)$ and $\epsilon_{\sigma_2} \sim \mathcal{N}(1.2, 0.1)$. Additionally, an extra source of variability for group 2 was injected 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:

$$\begin{aligned} x(t) &= \mathcal{N}(t; \mu_1 + \epsilon_{\mu_1}, \sigma_1 + \epsilon_{\sigma_1}) \\ &= \frac{1}{(\sigma_1 + \epsilon_{\sigma_1})\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(t - (\mu_1 + \epsilon_{\mu_1}))^2}{(\sigma_1 + \epsilon_{\sigma_1})^2}\right), \end{aligned} \tag{17}$$

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

$$\begin{aligned} x(t) &= \mathcal{N}(t; \mu_2 + \epsilon_{\mu_2}, \sigma_2 + \epsilon_{\sigma_2}) + \epsilon_a \\ &= \frac{1}{(\sigma_2 + \epsilon_{\sigma_2})\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(t - (\mu_2 + \epsilon_{\mu_2}))^2}{(\sigma_2 + \epsilon_{\sigma_2})^2}\right) + \epsilon_a \end{aligned} \tag{18}$$

Given the Gaussian density function $\mathcal{N}(t; \mu, \sigma)$, its 1st and 2nd derivative functions have two and three stationary points which are $\{\mu - \sigma, \mu + \sigma\}$ and $\{\mu - \sqrt{3}\sigma, 0, \mu + \sqrt{3}\sigma\}$, respectively. In our perspective, this suggests that the derivatives provide views that can exhibit additional discriminative features for pattern recognition purposes. A sample of 500 curves for each group and respective mean vectors are provided in Figure 1. From this illustration, note that larger distances between mean curves of group 1 and group 2 are observed around the stationary points.

For simulated data, as it is possible to analytically determine derivative functions and obtain a fine grained representation of the curves, there is no need to apply the spline smoothing procedure. Therefore, for all curves and derivatives, we computed their exact values on a grid from -4 to 4 with a 0.025 step length. Consequently, in this case, step 1 in Algorithm 1 is skipped.

We conducted preliminary tests with values in $\{0.1, 1, 10\}$ for the hyperparameter σ^s in the Gaussian kernel (15) and found that setting $\sigma^s = 1$

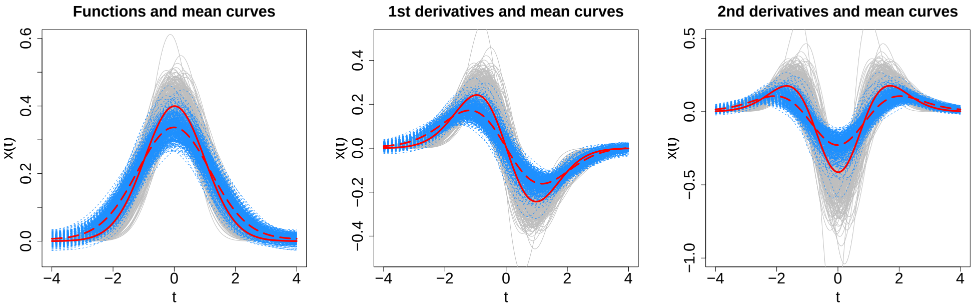

Figure 1: From left to right: original functions, 1st derivatives and 2nd derivatives. Functions from group 1 are in gray and solid lines, functions from group 2 are in blue and dashed lines. Mean functions are in red: thick solid lines correspond to group 1 whereas thick dashed lines represent group 2.

for all $s = 0, 1, 2$ yielded satisfactory results. Therefore, we used this value consistently across all experiments with artificial datasets. Moreover, since our focus is on comparing Gaussian kernel based representations with different sets of derivative functions, further tuning of this hyper-parameter was deemed unnecessary.

Tests with one sample of synthetic data. In the first set of experiments, we utilized the set of 1000 random curves represented in Figure 1. The clustering scores we obtained using linear and Gaussian kernels are illustrated in Figure 4. The box plots represent the variability of the NMI measures due to the kernel k-means initialization using 50 different random partitions.

In order to assess the different representations of functions with derivatives, we compared the NMI mean values (shown by red triangles in Figures). We used paired t-test with a significance level of 5% to determine whether the differences were statistically significant. To facilitate the readability of our conclusions, we use the following symbols $\sim$, $\gtrsim$, $>$ and $\gg$ for "similar to", "slightly better[4] than", "better[5] than", and "much better[6] than", respectively. It is important to note that $\sim$ indicates that the two representations are not statistically distinct.

[4] If the difference in absolute value is less than or equal than 5 points.

[5] If the difference in absolute value is between 5 and 10 points.

[6] If the difference in absolute value is greater than 10 points.

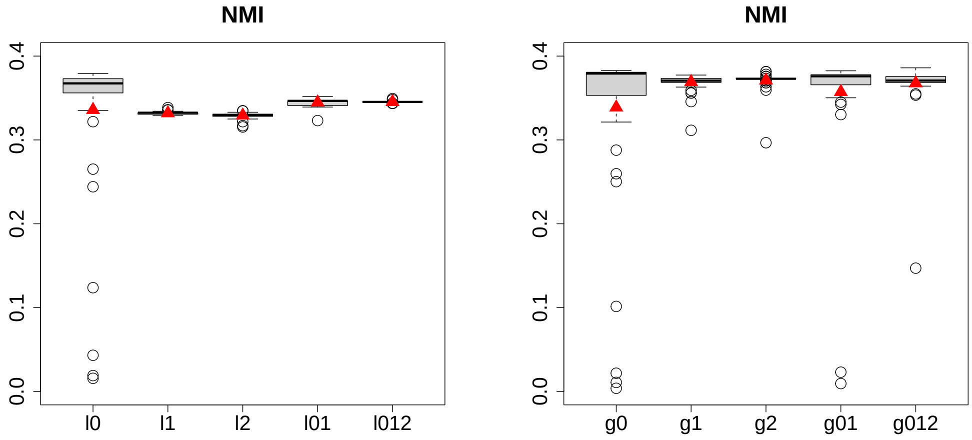

Figure 2: Box plots of NMI measures of MK-KM-FD using 50 random initializations for the kernel k-means procedure. Each box plot corresponds to a kernel representation with acronym given in x-axis (no weights optimization is applied). From left to right: linear kernel based representations then Gaussian kernel based representations. The red triangles indicate the mean values.

Regarding the linear representations based on the linear kernels, the graph on the left side of Figure 4 can be summarized as follows:

- $l0 \sim l1 \sim l2$.
- $l01 \gtrsim l1$ and $l01 \gtrsim l2$.
- $l012 \gtrsim l1$ and $l012 \gtrsim l2$.
- $l01 \sim l012$.

In the case of linear kernels, the different single views were comparable to each other but uniformly adding derivatives of different orders slightly improved the results.

In regard to FD representations using Gaussian kernels as a base similarity measure, the outcomes depicted in the graph on the right hand side of Figure 4 are different:

- $g1 \gtrsim g0$ and $g2 \gtrsim g0$.
- $g2 \gtrsim g1$.

- $g2 \sim g01 \sim g012$.

Here, the derivatives yielded slightly better NMI values compared to the original curves. In particular, 2nd derivatives provided the best performances. However, aggregating all information in a uniform multiview achieved comparable scores.

Next, we compare $l01$, $l012$, with $g01$, $g012$, which are the best FD representations we obtained so far with linear and Gaussian kernels. The NMI mean values[7] for these four approaches are 0.3448, 0.3453, 0.3572 and 0.3676, respectively. These measures suggest that Gaussian kernels outperform linear kernels, but only the following relations are found to be statistically significant:

- $g012 \gtrsim l01$ and $g012 \gtrsim l012$.

Here, we can conclude that both (i) integrating derivatives as complementary views of curves and (ii) employing non-linear kernels, can improve unsupervised pattern recognition.

Then, we studied the weights optimization procedure given in step 8 of Algorithm 1. We experimented with the same data and setting as before. Figure[8] 3 presents the comparison of previous multiview representations $l01t$, $l012t$, $g01t$ and $g012t$, without ($t = \emptyset$) and with weights optimization ($t = o$).

Weights optimization did not provide improvements in the case of FD representations using linear kernels:

- $l01 \gtrsim l01o$.
- $l012 \gtrsim l012o$.

In contrast, for Gaussien kernels, the NMI mean scores are higher when the views' weights are optimized. However, no statistical difference is achieved between $g01$ and $g01o$, or between $g012$ and $g012o$:

[7] As for illustration, the Purity measures for these four approaches are 0.8180, 0.8059, 0.8274 and 0.8392, respectively. In the case of $g012$, it means that, on average, 83.92% of the members of a found cluster are from a same group.

[8] For readability reasons, in Figure 3, we represented the NMI values that are in $[0.32, 0.385]$ only. Consequently, 3 low measures (below 0.32) from $g01$ and $g012$ are not represented.

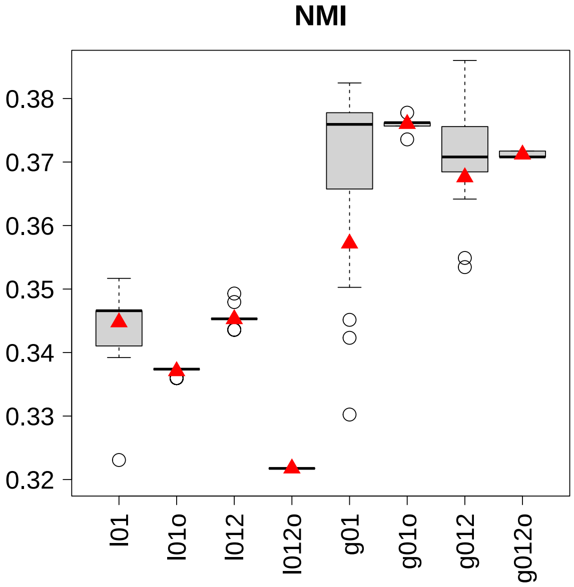

Figure 3: Box plots of NMI measures of MK-KM-FD using 50 random initializations for the kernel k-means procedure. Each box plot corresponds to a kernel representation with acronym given in x-axis. The acronym suffix o indicates weights optimization. From left to right: linear kernel based representations then Gaussian kernel based representations. The red triangles indicate the mean values.

- $g01 \sim g01o$.
- $g012 \sim g012o$.

Nonetheless, it is interesting to observe that, in these experiments, optimizing the weights helps reduce the variability of the clustering performances with respect to the random initialization of the kernel k-means. Indeed, in Figure 3, box plots of representations using weights optimizations are tighter compared to their respective counterparts.

Tests with 50 samples of synthetic data. In order to have a more global assessment of the MK-KM-FD approach, we generated 50 samples of the same kind of datasets as previously (500 curves per group). Our objective was to have a more robust evaluation of the impact of the weights optimization procedure. We applied the same setting as before, except for the number of random initializations of the kernel k-means which was reduced to 10. For each FD representation and each sample, we averaged the NMI measures over these 10 trials. The box plots of the distributions of NMI mean values with respect to the 50 samples are shown in Figure 4.

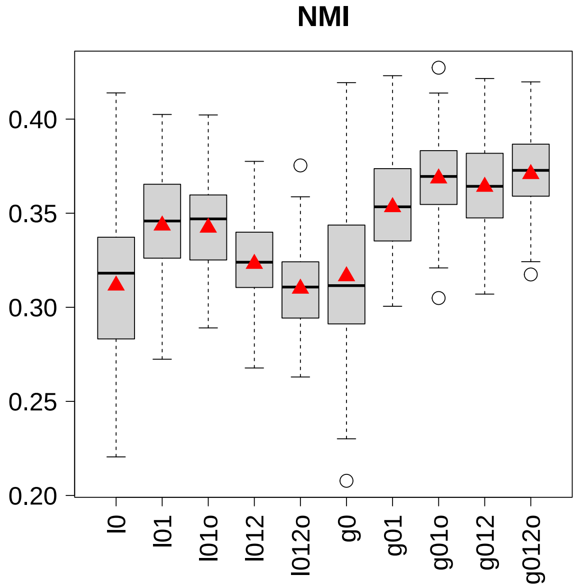

Figure 4: Box plots of NMI measures of MK-KM-FD using 50 samples of 1000 curves (half of group 1, half of group 2). Each box plot corresponds to a kernel representation with acronym given in x-axis. The acronym suffix o indicates weights optimization. From left to right: linear kernel based representations then Gaussian kernel based representations. The red triangles indicate the mean values.

These experiments confirm that Gaussian kernels generally yield better scores than linear kernels. Then, for each pair of representations based on Gaussian kernels, we statistically tested the difference in the distributions of the NMI mean values, using again a paired t-test with a significance level of 5%. The results were as follows:

- $g01o \gtrsim g01$.
- $g012o \gtrsim g012$.
- $g01o \sim g012o$.

Unlike the previous sample, these larger experiments demonstrate that, in the case of Gaussian kernels, weights optimization significantly enhance the clustering results, giving the best overall NMI scores.

In summary, these experimental outcomes clearly exemplify the potential benefits of our FD clustering technique MK-KM-FD, which promotes the use of derivative functions with non-linear kernel methods and a multiple views aggregation based on linear combination with optimized weights.

3.3.3. Experiments with real-world data

In addition to synthetic data, we examined 6 real-world datasets, the characteristics of which are exhibited in Table 1. All datasets are publicly available from either the *fda* R package [28], the *fda.usc* R package [29] or the UEA and UCR TS Classification Repository [30]. Here is a brief description of each dataset:

- *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 discriminate 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.

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.

Unlike simulated data, pre-processings are necessary to recover an approximated functional form from the available discrete observations. Following the procedure outlined in sub-section 2.4, we carried out the spline smoothing approach given by (8) with a roughness penalty $R(x) = \|D^4 x\|^2_{\mathbb{L}_2}$ for all 6 datasets. This pre-processing corresponds to step 1 of Algorithm 1.

We exclusively tested the Gaussien kernel based representations. For determining the neighborhood bandwidth $\sigma^s, \forall s = 0, 1, 2$, we employed a strategy inspired by [31] for auto-tuning this hyper-parameter. In their study, the authors proposed a local scaling approach where, for each pair $(x_i, x_{i'})$, $(\sigma^s)^2$ in (15) was replaced with $\sigma^s_i \sigma^s_{i'}$, where σ^s_i represents the distance from $D^s x_i$ to its 7th nearest neighbor. While this method proved effective, it may yield an affinity matrix that is not positive semi-definite. To circumvent this issue, we determined the empirical distribution of the distances to the 7th nearest neighbors for each dataset, and set the global σ^s value to the median estimate.

Similar to the simulated data experiments, we conducted 10 runs of MK-KM-FD 10 times with distinct random initializations. Figure 5 illustrates the distributions of the NMI scores for all 6 datasets considering representations $g0$, $g1$, $g2$, $g01$, $g01o$, $g012$, and $g012o$.

We begin with a summary of the performances provided by the single views $g0$, $g1$ and $g2$. Below are ranked lists of representations organized in descending order of the NMI mean values (depicted as red triangles in Figure 5). Then, for two subsequent representations, we indicate a preference

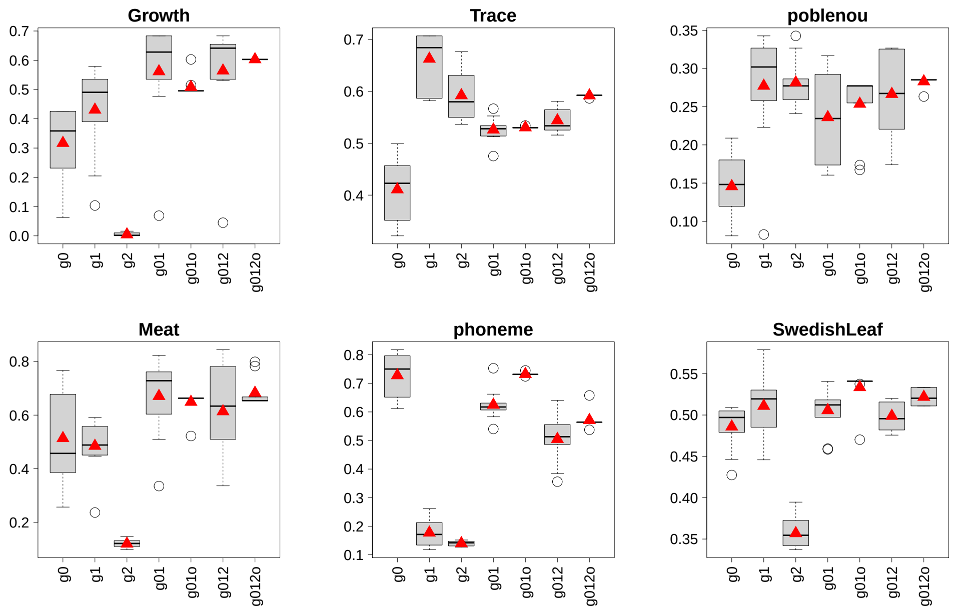

Figure 5: Box plots of NMI measures of MK-KM-FD using 10 random initializations for the kernel k-means procedure. Each box plot corresponds to a Gaussian kernel based representation with acronym given in x-axis. The acronym suffix o indicates weights optimization. Each graphic corresponds to the results of a real-world dataset whose name is specified on top of the frame. The red triangles indicate the mean values.

relation ($\gtrsim$ of $>$ or $\gg$) only if it is statistically significant[9] otherwise we use $\sim$:

- *Growth*: $g1 \sim g0 \gg g2$.
- *Trace*: $g1 > g2 \gg g0$.
- *poblenou*: $g2 \sim g1 \gg g0$.
- *Meat*: $g0 \sim g1 \gg g2$.
- *phoneme*: $g0 \gg g1 \gtrsim g2$.

[9] Note that the relation "significantly different" provided by the paired t-test is not transitive.

- *SwedishLeaf*: $g1 \sim g0 \gg g2$.

While derivatives can outperform the original functions, it is generally uncertain in advance which order of derivation to prioritize. Hence, it is pertinent to explore the multiview approach and enhance the comparisons above by incorporating $g01$ and $g012$. When contrasting the most effective and least effective single views with the multiview models, we find:

- *Growth*: $g012 \sim g01 \gg g1 \gg g2$.
- *Trace*: $g1 \gg g012 \sim g01 \gg g0$.
- *poblenou*: $g2 \sim g012 \sim g01 > g0$.
- *Meat*: $g01 \sim g012 > g0 \gg g2$.
- *phoneme*: $g0 \gg g01 \gg g012 \gg g2$.
- *SwedishLeaf*: $g1 \sim g01 \sim g012 \gg g2$.

We highlight that multiview representations offer risk-averse strategies. While they may not always deliver the best clustering performance, they consistently avoid the worst case scenario.

Next, we complete our analysis by investigating whether optimizing the views' weights enhances clustering performance:

- *Growth*: $g012o \sim g012 \sim g01 \sim g01o$.
- *Trace*: $g012o \gtrsim g012 \sim g01o \sim g01$.
- *poblenou*: $g012o \sim g012 \sim g01o \sim g01$.
- *Meat*: $g012o \sim g01 \sim g01o \sim g012$.
- *phoneme*: $g01o \gg g01 > g012o > g012$.
- *SwedishLeaf*: $g01o \sim g012o \sim g01 \sim g012$.

In the vast majority of cases, optimizing the views' weights provided better outcomes. Furthermore, similar to the experiments with synthetic data, we observed from Figure 5 that the dispersion of the NMI values was tighter when weights are optimized, suggesting that MK-KM-FD is less dependant on the random initialization of the kernel k-means procedure.

In summary, we argue that, without any external knowledge on the FD, using Gaussian kernel functions, setting q = 2 and conducting weights optimization could be advantageous. This configuration makes it possible to (i) mitigate the risks associated with uncertainty regarding the choice of views and the manifold hypothesis, and (ii) potentially achieving superior clustering performances.

4. Multiple kernel classification of functions with derivatives

Henceforth, we shift our focus to the classification task. As in section 3, we begin with a short review of related works in the scope of FD classification. Then, we delve into the details of our technique which relies on multiple kernel SVM. Once again, each set of derivative functions $\{D^s x_i\}_i$ with $s = 0, \ldots, \mathrm{q}$, is seen as a distinct view of the objects under study. Leveraging kernel functions enables us to adopt a more versatile representation framework capable of handling non-linear manifolds. Proposition 1 is also employed in this supervised setting to provide a principled approach for updating the weights of the different views. Our experiments use the same synthetic and real data as in the preceding section.

4.1. Previous works on functional data classification

From a broad perspective, let $\mathbb{X}$ be the data space, $\mathbb{C}$ be the discrete and finite label set, $c_i \in \mathbb{C}$ represent the class of $x_i \in \mathbb{X}$, and $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$ be the training set. In the classification task, the goal is to learn from $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$, a mapping $f : \mathbb{X} \to \mathbb{C}$ that accurately predicts $c \in \mathbb{C}$ for any given $x \in \mathbb{X}$.

There are numerous classification techniques available. In this paper, we focus on parametric models where the induction phase involves selecting an appropriate instance from a class of functions by minimizing a loss function. In this context, several classic multivariate methods have been extended to FD including Linear Discriminant Analysis (LDA) [32], Quadratic Discriminant Analysis (QDA) [33, 34], logistic regression and more globally, generalized linear models [35, 36].

Predictive methods from the machine learning community have also inspired researchers and practitioners working with FD. For instance, functional random forest approaches were introduced in studies such as [37] and [38]. Neural networks and ensemble methods are other machine learning techniques that have been explored as seen in [39, 40, 41] and [42, 12] respectively.

In this paper, we are interested in kernel methods. Firstly, we mention [43], which proposes projecting FD into a RKHS for binary classification problems using a penalized logistic regression as a prediction model. Another relevant work is [44], which extends Support Vector Machines (SVM) to FD. The authors highlight transformations suitable for dealing with FD and deriving meaningful kernels. Moreover, [44] establishes the consistency of the functional SVM algorithm following the reasoning introduced in [45].

The SVM method for FD mentioned above serves as the foundation of our classification model for FD with derivatives. Our contribution can be viewed as an extension of [44], where we explicitly consider functions in $\mathbb{H}^{\mathrm{q}}$ instead of $\mathbb{L}^2$, and we apply the multiple kernel learning framework to combine the kernel matrices associated with distinct sets of derivatives. Further details are provided in the following sub-section.

4.2. The multiple kernel SVM model for FD with derivatives

To begin with, let us recall the functional SVM model introduced in [44] in a more formal manner. In this latter paper, the FD $\{x_i\}_i$ are considered elements of $\mathbb{L}^2([0,T])$. Given a training set $\{(x_i, c_i)\}_{i=1,\ldots,\mathrm{n}}$, the SVM approach for FD consists in solving the following convex optimization problem (primal):

$$\begin{aligned} &\min_{a_0\in\mathbb{R}, a\in\mathbb{L}^2} \frac{1}{2}\|a\|^2_{\mathbb{L}^2} + \mu\sum_{i=1}^{\mathrm{n}} \xi_i \qquad (19)\\ &\text{s.t.} \begin{cases} c_i\left(a_0 + \langle a, x_i\rangle_{\mathbb{L}^2}\right) \geq 1 - \xi_i, \forall i = 1,\ldots,\mathrm{n};\\ \xi_i \geq 0, \forall i = 1,\ldots,\mathrm{n}, \end{cases}\end{aligned}$$

where $\mu \geq 0$ is a hyper-parameter regulating 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:

$$\min_{a_0\in\mathbb{R}, a\in\mathbb{L}^2} \frac{1}{2}\|a\|^2_{\mathbb{L}^2} + \mu\sum_{i=1}^{\mathrm{n}} \max\left(0, 1 - c_i\left(a_0 + \langle a, x_i\rangle_{\mathbb{L}^2}\right)\right) \qquad (20)$$

The SVM methodology has a notable characteristic, namely its dual formulation, which is expressed as follows:

$$\begin{aligned}&\max_{\boldsymbol{\alpha}\in\mathbb{R}^{\mathrm{n}}}\sum_{i=1}^{\mathrm{n}}\alpha_i-\frac{1}{2}\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}}\alpha_i\alpha_{i'}c_ic_{i'}\langle x_i,x_{i'}\rangle_{\mathbb{L}^2}\\ &\text{s.t. }\left\{\begin{array}{l}\sum_{i=1}^{\mathrm{n}}\alpha_ic_i=0;\\ 0\leq\alpha_i\leq\mu,\forall i=1,\ldots,\mathrm{n}.\end{array}\right.\end{aligned} \tag{21}$$

The duality transforms the primal problem, whose solution is in the function space $\mathbb{L}^2$, into a dual problem with a finite dimensional search space $\mathbb{R}^{\mathrm{n}}$. Furthermore, the dual problem solely depends on the inner products between pairs of objects in the training sample. This property makes it possible to implicitly project the FD into a RKHS using 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 mapping. Then, the SVM approach in its dual expression can be formulated as follows:

$$\begin{aligned}&\max_{\boldsymbol{\alpha}\in\mathbb{R}^{\mathrm{n}}}\sum_{i=1}^{\mathrm{n}}\alpha_i-\frac{1}{2}\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}}\alpha_i\alpha_{i'}c_ic_{i'}\mathbf{K}_{ii'}\\ &\text{s.t. }\left\{\begin{array}{l}\sum_{i=1}^{\mathrm{n}}\alpha_ic_i=0;\\ 0\leq\alpha_i\leq\mu,\forall i=1,\ldots,\mathrm{n}.\end{array}\right.\end{aligned} \tag{22}$$

We extend the previous SVM model for functions in $\mathbb{L}^2$ to incorporate the multiple kernel SVM approach applied to functions in the Sobolev space $\mathbb{H}^{\mathrm{q}}$. Following our general framework described in section 2.2, we propose to employ a multiple kernel matrix:

$$\mathbf{K}=\sum_{s=0}^{\mathrm{q}}w_s\mathbf{K}^s, \tag{23}$$

where $w_s\geq 0,\forall s=0,\ldots,\mathrm{q}$, and for all couples $(x_i,x_{i'})$ in the training set, $\mathbf{K}^s_{ii'}=\langle\psi^s(D^sx_i),\psi^s(D^sx_{i'})\rangle_{\mathbb{F}^s}=k^s(D^sx_i,D^sx_{i'})$ similar to the materials exposed in sub-section 3.2.

As in the unsupervised case, we leverage the fact that the derivative functions offer various perspectives of the original objects and employ the multiple kernel learning paradigm to merge these distinct sources of information. Moreover, we project each set $\{D^sx_i\}_i$ for all $s=0,\ldots,\mathrm{q}$, from $\mathbb{L}^2$

to an RKHS using the mapping functions ψ^s. This aspect becomes crucial when classes are not linearly separable.

In the supervised case as well, we suppose that the kernel matrices $\{\mathbf{K}^s\}_s$ should complement each other rather than compete with each other. Therefore, we are in line with the general approach studied in [46, 17] which promotes the constraint $\|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1$ with $\mathrm{r} > 1$. Essentially, our method can be viewed as an extension of the latter model from vectors in $\mathbb{R}^\mathrm{p}$ to functions with derivatives in $\mathbb{H}^\mathrm{q}$. We aim to solve the following problem:

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathrm{q}+1}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^\mathrm{n}} \sum_{i=1}^{\mathrm{n}} \alpha_i - \frac{1}{2}\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}} \alpha_i\alpha_{i'}c_ic_{i'}\sum_{s=0}^{\mathrm{q}} w_s\mathbf{K}^s_{ii'} \tag{24}$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{\mathrm{n}} \alpha_i c_i = 0; \\ 0 \leq \alpha_i \leq \mu, \forall i = 1,\ldots,\mathrm{n}; \\ \mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1. \end{cases}$$

The optimization procedure is similar to the unsupervised case and consists in alternating between (i) maximizing with respect to $\boldsymbol{\alpha}$ with a fixed $\mathbf{w}$ using the regular SVM algorithm, and (ii) minimizing with respect to $\mathbf{w}$ with a fixed $\boldsymbol{\alpha}$. The second problem has a closed-form solution that can be stated using Proposition 1. Let us consider the opposite of the minimization in $\mathbf{w}$, and introduce the vector $\mathbf{z} \in \mathbb{R}^{\mathrm{q}+1}$ with elements given by:

$$z_s = \sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}} \alpha_i\alpha_{i'}c_ic_{i'}\mathbf{K}^s_{ii'}, \quad \forall s = 0,\ldots,\mathrm{q}. \tag{25}$$

Note that since for all s, $\mathbf{K}^s$ is positive semi-definite, hence z_s is non-negative.

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

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathrm{q}+1}} \sum_{i=1}^{\mathrm{n}} \alpha_i - \frac{1}{2}\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}} \alpha_i\alpha_{i'}c_ic_{i'}\sum_{s=0}^{\mathrm{q}} w_s\mathbf{K}^s_{ii'} \tag{26}$$
$$s.t. \begin{cases} \mathbf{w} \geq \mathbf{0} \\ \|\mathbf{w}\|_{\ell_\mathrm{r}} \leq 1, \end{cases}$$

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

$$w_s^* = \frac{\left(\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}} \alpha_i\alpha_{i'}c_ic_{i'}\mathbf{K}^s_{ii'}\right)^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=0}^{\mathrm{q}}\left(\sum_{i=1}^{\mathrm{n}}\sum_{i'=1}^{\mathrm{n}} \alpha_i\alpha_{i'}c_ic_{i'}\mathbf{K}^{s'}_{ii'}\right)^{\frac{\mathrm{r}}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}}}. \tag{27}$$

Algorithm 2: Multiple kernel SVM for functions with derivatives (MK-SVM-FD).

```
Input: ${y_{ij}}_{i=1,\dots,\mathrm{n};j=1,\dots,p}$ (sampled values of FD), $\mathrm{q} \geq 0$ (maximum
       order of derivative), $\mathrm{r} > 1$ ($\ell_\mathrm{r}$ norm, default 2), $\{k^s\}_{s=0,\dots,\mathrm{q}}$
       (kernel functions, default Gaussian), $\sigma$ (kernel
       hyper-parameter if any)
Output: $\boldsymbol{\alpha}$ (support vectors' 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 $\{x_i\}_{i=1,\dots,\mathrm{n}}$ by solving (8);
Determine $\{D^s x_i\}_{i=1,\dots,\mathrm{n}}, \forall s = 1, \dots, \mathrm{q}$;
Determine $\{\mathbf{K}^s = (k^s(D^s x_i, D^s x_{i'}))_{i,i'=1,\dots,\mathrm{n}}\}, \forall s = 0, \dots, \mathrm{q}$;
Normalize the kernel matrices $\mathbf{K}^s, \forall s = 0, \dots, \mathrm{q}$ (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
```

In Algorithm 2 we give the pseudo-code of our multiple kernel SVM procedure for functions with derivatives that we denote by MK-SVM-FD. Likewise the clustering case, the overall objective function is improved at each iteration, as a consequence Algorithm 2 converges to a local optimum.

4.3. Experiments with the MK-SVM model for functions with derivatives

4.3.1. Experiments settings

We are interested in exploring the same research questions raised in sub-section 3.3, but within a supervised context. Consequently, we now incorporate the labels of the data into the learning process to infer mappings aimed at predicting the correct label c given any instance $x \in \mathbb{H}^{\mathrm{q}}$. Our objective is to compare the prediction functions derived from different FD representations using various sets of derivatives, kernel functions, and weighting schemes for the views. Accordingly, similar to sub-section 3.3.1, we examined the different kernel matrices $\mathbf{K}^{a0}$, $\mathbf{K}^{a1}$, $\mathbf{K}^{a2}$, $\mathbf{K}^{a01t}$ and $\mathbf{K}^{a012t}$ with $a = l, g$ and $t = \emptyset, o$.

In the supervised learning case, the evaluation criterion we used is the accuracy rate, given by:

$$\text{Accuracy}(C, L) = \frac{1}{\mathrm{n}} \sum_{l=1}^{\mathrm{k}} |C_l \cap L_l| \tag{28}$$

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

4.3.2. Experiments with simulated data

The artificial data used for the supervised task is the same as for the unsupervised task as detailed in sub-section 3.3.2. We randomly generated 1200 curves with 600 coming from group 1 given by (18), and 600 from group 2 following (19). The sample was randomly split into a training set of 1000 curves and a test set of 200 curves. In regard to the hyper-parameter μ, it was estimated in a grid search manner and chosen among the following values: $\{0.01, 0.1, 1, 10, 100\}$. More precisely, we compared the performances of each value based on a 5-fold cross-validation using the training set. The value of μ that provided the best validation accuracy rate on average was selected. We then estimated the model again using this tuned hyper-parameter value but utilizing the entire training set. Subsequently, we applied the latter model on the test set of 200 curves and measured the (test) accuracy rate.

We repeated this procedure 50 times. The variations of the test accuracy rates are shown in Figure 6 for both linear and Gaussian kernel based representations.

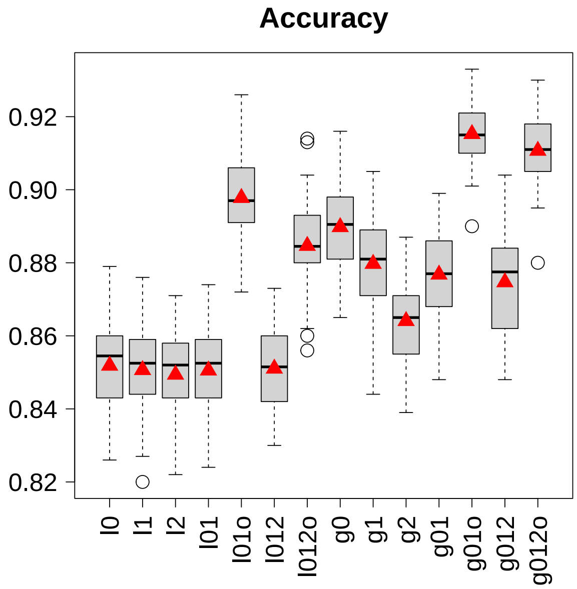

Figure 6: Box plots of test accuracy rates of MK-SVM-FD using 50 samples. 1000 curves (half of group 1, half of group 2) were used for training and 200 curves for testing. Each box plot corresponds to a kernel representation with acronym given in x-axis. The acronym suffix o indicates weights optimization. From left to right: linear kernel based representations then Gaussian kernel based representations. The red triangles indicate the mean values.

As before, we used paired t-test with a significance level of 5% in order to provide a more robust comparison of the assessment scores between representations. If the null hypothesis of no difference between the mean averages is accepted, it is indicated with the symbol $\sim$.

Figure 6 demonstrates the superiority of Gaussian based representations over the linear based ones. We can summarize these findings as follows:

- For single views: $gs \gtrsim ls$ with $s = 0, 1, 2$.
- For multiple views: $gst \gtrsim lst$ with $s = 01, 012$, and $t = \emptyset, o$.

Using derivatives and uniform weights did not boost the performances:

- $g0 \gtrsim g01$,
- $g0 \gtrsim g012$,
- $g01 \sim g012$.

However, weights optimization did improve the results:

- $g01o > g01$ and $g01o > g0$,
- $g012o > g012$ and $g012o > g0$.

In summary, for the artificial dataset, Gaussian kernel based representations outperform linear kernel based representations. Furthermore, incorporating derivatives increases the assessment scores but only when weights optimization is carried out. The two best models are given by the representations $g01o$ and $g012o$ which support our classification model based on a multiple kernel framework with optimized weights.

4.3.3. Experiments with real-world data

Next, we applied MK-SVM-FD to the 6 real-world datasets described previously in Table 1. In the multiclass case, we applied SVM using a one-versus-one strategy[10] and a voting scheme for prediction. To determine the optimal hyper-parameter μ, we conducted a 10-fold cross-validation. The grid search was performed within the subset $\{0.01, 0.1, 1, 10, 100\}$. The selected value was the one that resulted in the highest average validation accuracy rate over the 10 folds. Since these datasets are small, we did not utilize a separate test set. Therefore, our analysis is based on the mean validation accuracy rates across the 10 folds.

In most cases, the Gaussian kernel based representations provided better assessment measures than the linear kernel based representations. Therefore, in what follows, we only expose the experimental results related to the use of the Gaussian kernel. Figure 7 illustrates the box plots representing the distributions of the validation accuracy rates of the tuned model for each representation among $g0$, $g1$, $g2$, $g01$, $g01o$, $g012$ and $g012o$.

For each dataset, we provide below the ranked list of single view representations in descending order of the mean validation accuracy rates. For two

[10] In this case, the overall objective function is the sum of objective functions of binary classifiers across all pairs of classes.

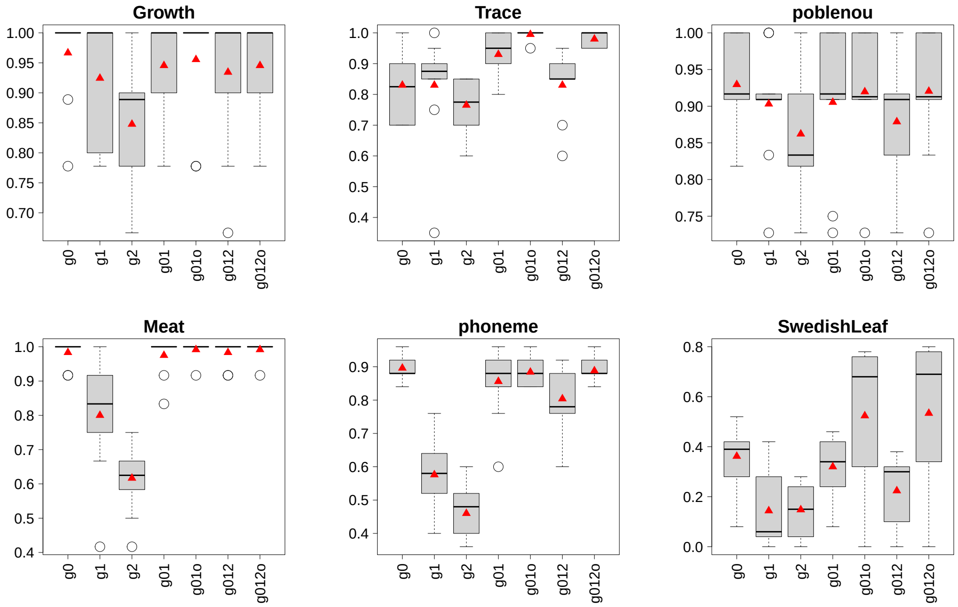

Figure 7: Box plots of validation accuracy rate measures of MK-SVM-FD using 10-fold cross-validation. Each box plot corresponds to a Gaussian kernel based representation with acronym given in x-axis. The acronym suffix o indicates weights optimization. Each graphic corresponds to the results of a real-world dataset whose name is specified on top of the frame. The red triangles indicate the mean values.

subsequent cases, we compared the distributions of the validation accuracy rates over the folds using the same paired t-test as before. We obtained the following outcomes:

- *Growth*: $g0 \gtrsim g1 > g2$.
- *Trace*: $g0 \sim g1 > g2$.
- *poblenou*: $g0 \sim g1 \sim g2$.
- *Meat*: $g0 \gg g1 \gg g2$.
- *phoneme*: $g0 \gg g1 \gg g2$.
- *SwedishLeaf*: $g0 \gg g2 \sim g1$.

For all 6 cases, original functions provided the best scores compared to 1st order and 2nd order derivative functions.

Next, we add the results of multiview representations with uniform weights. The relative positions of $g01$ and $g012$ with respect to the best and worst single view performances are as follows:

- *Growth*: $g0 \sim g01 \sim g012 \gg g2$.
- *Trace*: $g01 > g012 \sim g0 > g2$.
- *poblenou*: $g0 \sim g01 \sim g012 \sim g2$.
- *Meat*: $g012 \sim g0 \sim g01 \gg g2$.
- *phoneme*: $g0 \sim g01 > g012 \gg g2$.
- *SwedishLeaf*: $g0 \sim g01 \gtrsim g012 \sim g1$.

Despite differences in mean validation accuracy scores, the multiview approach $g01$ performed similarly to or better than the best single view, $g0$. More broadly, as with the clustering task, multiple kernel representations are generally risk-averse strategies for classification problems: the accuracy scores of multiview representations are always higher than those of the least performing single view.

Considering non-uniform metrics between functions and derivatives using weights optimization, we got the following ranked lists:

- *Growth*: $g01o \sim g012o \sim g01 \sim g012$.
- *Trace*: $g01o \sim g012o \sim g01 > g012$.
- *poblenou*: $g012o \sim g01o \sim g01 \sim g012$.
- *Meat*: $g012o \sim g01o \sim g012 \sim g01$.
- *phoneme*: $g012o \sim g01o \sim g01 > g012$.
- *SwedishLeaf*: $g012o \sim g01o \gg g01 \gtrsim g012$.

Based on these benchmarks, we can conclude that weights optimization is a much better strategy than using uniform weights when integrating derivatives. In the *Trace* and *SwedishLeaf* cases in particular, weights optimization dramatically boosts accuracy scores.

The summary of the experimental results on classification tasks suggests that Gaussian kernels generally outperform linear kernels. Moreover, incorporating derivatives and optimizing the views' weights perform as well as or better than the best single view representation. These outcomes support our MK-SVM-FD method.

5. Conclusion and future work

FDA offers a powerful means to analyze continuous phenomena through statistical and machine learning techniques, enriched with tools from functional analysis. In this paper, we considered FD as elements of the Sobolev space $\mathbb{H}^q$. We have presented a new framework enabling the learning of versatile representations of FD for both clustering and classification purposes. Our approach relies on two key components. Firstly, we leverage kernel methods to implicitly map FD and their derivative functions into (potentially distinct) RKHS. Secondly, we introduce methods to learn how to combine these kernel functions for both unsupervised (MK-KM-FD) and supervised (MK-SVM-FD) learning tasks.

In our experimental evaluations using both simulated and real-world data, we observed that employing a Gaussian kernel can significantly enhance both clustering and classification performances compared to using a linear kernel. Additionally, we found that optimizing the weights further improve clustering outcomes and reduces sensitivity to random initialization in the clustering task. In classification, MK-SVM-FD with weights optimization delivered remarkable performances. Overall, our methods demonstrate robustness in combining multiple views provided by successive derivatives, making them well-suited for clustering and classifying smooth functional data, especially when the quality of individual representations is uncertain.

In future work, we plan to extend our framework by incorporating weights functions instead of scalar weights to balance each derivative order. This approach could leverage techniques from sparse clustering and interpretable SVM for FD. Furthermore, our methods can be extended to handle multivariate functional data, which naturally pose multiview learning problems. Another promising direction is exploring physics-informed machine learning, where we aim to integrate more general differential operators into our framework to incorporate physics constraints into the learning process.

Author contributions

Julien Ah-Pine: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - Original Draft, Writing - Review & Editing.
Anne-Françoise Yao: Conceptualization, Writing - Review & Editing.

Acknowledgment

The authors express their gratitude to the anonymous referees for their valuable comments, which have contributed to the improvement of the paper. 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.

Proof of Proposition 1

Proof. The Lagrangian function of Problem (3) reads:

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

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:

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

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

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

By plugging (C) into (B) we obtain the following stationary point which states Equation (4) of Proposition 1:

$$\mathbf{w}^* = (w_s^*)_{s=0,\ldots,\mathrm{q}} \text{ with } w_s^* = \frac{z_s^{\frac{1}{\mathrm{r}-1}}}{\left(\sum_{s'=0}^{\mathrm{q}} z_{s'}^{\frac{\mathrm{r}}{\mathrm{r}-1}}\right)^{\frac{1}{\mathrm{r}}}}.$$

Moreover, the KKT multipliers are given by:

$$\begin{aligned}\boldsymbol{\alpha}^* &= \mathbf{0}_{\mathrm{q}+1}, \\ \beta^* &= \left(\sum_{s=0}^{\mathrm{q}} z_s{}^{\frac{r}{\mathrm{r}-1}}\right)^{\frac{\mathrm{r}-1}{\mathrm{r}}} = \|\mathbf{z}\|_{\ell_{\mathrm{r}/(\mathrm{r}-1)}}.\end{aligned}$$

Next, we need to prove that $(\mathbf{w}^*, \boldsymbol{\alpha}^*, \beta^*)$ is a maximizer. To this end, we need to study $\nabla^2_\mathbf{w} L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \beta^*)$, the Hessian matrix of the Lagrangian function with respect to $\mathbf{w}$ evaluated at $(\mathbf{w}^*, \boldsymbol{\alpha}^*, \beta^*)$. From (A) we can see that $\nabla^2_\mathbf{w} L(\mathbf{w}, \boldsymbol{\alpha}, \beta)$ is the same as $\nabla^2_\mathbf{w} M(\mathbf{w}, \beta)$ with:

$$M(\mathbf{w}, \beta) = -\beta \|\mathbf{w}\|_{\ell_\mathrm{r}}.$$

By the Minkowski inequality, we can easily show that $\|\mathbf{w}\|_{\ell_\mathrm{r}}$ is a strictly convex function for $\mathrm{r} > 1$. Furthermore, since $\beta^* > 0$, we deduce that $\nabla^2_\mathbf{w} M(\mathbf{w}^*, \beta^*)$ is negative definite and so is $\nabla^2_\mathbf{w} L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \beta^*)$. As a consequence, the second order sufficient conditions are met and $\mathbf{w}^*$ is a global maximizer. □

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve the language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

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

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

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

[4] E. Keogh, M. Pazzani, Dynamic time warping with higher order features, in: Proceedings of the 2001 SIAM Intl. Conf. on Data Mining, Vol. 2, 2001.

[5] T. Górecki, M. Łuczak, Using derivatives in time series classification, Data Mining and Knowledge Discovery 26 (2013) 310–331.

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

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

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

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

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

[16] J. C. Bezdek, R. Ehrlich, W. Full, Fcm: The fuzzy c-means clustering algorithm, Computers & Geosciences 10 (2-3) (1984) 191–203.

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

[18] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, Lp-norm multiple kernel learning, The Journal of Machine Learning Research 12 (2011) 953–997.

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

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

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

[22] 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.

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

[24] 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.

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

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

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

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

[29] 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/

[30] 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.

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

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

[33] 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.

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

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

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

[37] 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.

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

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

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

[41] 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.

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

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

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

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

[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.