K-means algorithms for functional data

María Luz López García [a], Ricardo García-Ródenas [a,*], Antonia González Gómez [b]

[a] Departamento de Matemáticas, Escuela Superior de Informática, Universidad de Castilla la Mancha, 28012 Ciudad Real, Spain
[b] Departamento de Matemática Aplicada a los Recursos Naturales, E.T. Superior de Ingenieros de Montes, Universidad Politécnica de Madrid, 28040 Madrid, Spain

ARTICLE INFO

Keywords:
Functional data
K-means
Reproducing Kernel Hilbert Space
Tikhonov regularization theory
Dimensionality reduction

ABSTRACT

Cluster analysis of functional data considers that the objects on which you want to perform a taxonomy are functions $f : X \subset \mathbb{R}^p \mapsto \mathbb{R}$ and the available information about each object is a sample in a finite set of points $f_n = \{(x_i, y_i) \in X \times \mathbb{R}\}_{i=1}^n$. The aim is to infer the meaningful groups by working explicitly with its infinite-dimensional nature.

In this paper the use of *K*-means algorithms to solve this problem is analysed. A comparative study of three *K*-means algorithms has been conducted. The *K*-means algorithm for raw data, a kernel *K*-means algorithm for raw data and a *K*-means algorithm using two distances for functional data are tested. These distances, called d_{V_n} and d_ϕ, are based on projections onto Reproducing Kernel Hilbert Spaces (RKHS) and Tikhonov regularization theory. Although it is shown that both distances are equivalent, they lead to two different strategies to reduce the dimensionality of the data. In the case of d_{V_n} distance the most suitable strategy is Johnson–Lindenstrauss random projections. The dimensionality reduction for d_ϕ is based on spectral methods.

A key aspect that has been analysed is the effect of the sampling $\{x_i\}_{i=1}^n$ on the *K*-means algorithm performance. In the numerical study an *ex professo* example is given to show that if the sampling is not uniform in *X*, then a *K*-means algorithm that ignores the functional nature of the data can reduce its performance. It is numerically shown that the original *K*-means algorithm and that suggested here lead to similar performance in the examples when *X* is uniformly sampled, but the computational cost when working with the original set of observations is higher than the *K*-means algorithms based on d_ϕ or d_{V_n}, as they use strategies to reduce the dimensionality of the data.

The numerical tests are completed with a case study to analyse what kind of problem the *K*-means algorithm for functional data must face.

1. Introduction

Cluster analysis allows data structures to be explored. These tools provide a first intuitive data structure by identifying meaningful groups. Two essential elements in cluster analysis are data representation and the specification of similarity between them. Most of these methods assume that the objects can be represented as points in Euclidean spaces $\mathbb{R}^n$, but for some problems data are random functions (physical processes, genetic data, chemical spectra, voice recording, control processes, etc). In this paper we assume that the objects are functions $f : X \subset \mathbb{R}^p \mapsto \mathbb{R}$ and the information provided by each of them is a sampling of a finite set of points $f_n = \{(x_i, y_i) \in X \times \mathbb{R}\}_{i=1}^n$. The aim is to infer the data structure by working explicitly with their infinite dimensional nature in Hilbert spaces. This field is known as *functional data analysis* (FDA) [29,11].

Jacques and Preda [17] establish a classification of the different clustering methods for functional data as follows: (i) raw-data clustering, (ii) two-stage methods, (iii) non-parametric clustering and (iv) model-based clustering. A similar classification for clustering of time-series data is proposed by Liao [22]. The methods of type (i) work directly with raw data, type (ii) indirectly with features extracted from the raw data, type (iii) use a specific distance or dissimilarity between functions and type (iv) with models built from the raw data, in which the estimation of features and clustering are performed simultaneously.

Methods (ii)–(iv) developed in functional data clustering use Hilbert spaces to tackle the functional nature of the data and to obtain a representation of these data. Many of the methods choose an orthogonal basis of functions $\Phi = \{\varphi_1, \ldots, \varphi_L\}$ with $L \in \mathbb{N}$ and each functional datum is represented as a linear combination of the vectors in the basis Φ. Usual choices of Φ are Fourier, *Wavelets* or *B*-splines. In this paper following [15], we consider each function as

* Corresponding author.
E-mail addresses: MariaLuz.Lopez@uclm.es (M. Luz López García), ricardo.garcia@uclm.es (R. García-Ródenas), antonia.gonzalez@upm.es (A. González Gómez).

a point in a general function space and then project these points onto *a Reproducing Kernel Hilbert Space* (RKHS) by using the Tikhonov regularization theory. This mechanism induces a distance among the functions of the sample and therefore it allows clustering algorithms to be applied to functional data. This process is completed with a strategy for dimensionality reduction consisting of a new projection onto a finite-dimensional Euclidean space which makes the distance between the functions coincide with the Euclidean distance of the projected data. The method proposed in this paper is an instance of a two-stage method but it allows the algorithms of this class to be reinterpreted (mainly based on it functional principal component analysis) as methods that compute the similarity measure as the distance between the projections of the data onto a Hilbert space.

The mathematical theory of RKHS [25] has been applied to several fields such as *support vector machines*(SVM) [9], *principal component analysis* [30], *canonical correlation analysis* [13] and *Fisher's discriminant analysis* [26]. RKHS has also been applied to finite-dimensional cluster analysis giving rise to the so-called kernel-based clustering methods. Filippone et al. [12] classify clustering methods based on kernels into three categories: (i) methods based on kernelization of the distance, (ii) clustering in *feature spaces* and (iii) methods based on SVM. It has been experimentally shown that methods based on kernels allow correct grouping of clusters with nonlinear borders.

This paper is focused on the *K*-means algorithm [24], which may have been the most popular clustering algorithm since the 60s. This algorithm has been deeply range-studied in the finite case [33] and it has been extended to many situations as *K*-means based on kernels, see [30,14,8].

Cadre and Paris [5] developed a numerical *K*-means algorithm for infinite-dimensional Hilbert spaces. This numerical scheme discretizes functions on grids and considers that all grid cells have the same volume. These authors show that the theoretical performance of this algorithm matches the classical.

Functional data clustering has also been addressed by the *K*-means algorithm in [1,28]. As a first step the dimension of the data is reduced, and then as a second step the *K*-means for finite dimensional data is used. In [1] the dimension reduction strategy consists of an approximation of the curves into a finite *B*-spline basis and the *B*-spline coefficients are used as feature vectors. Peng and Müller [28] use principal component scores to reduce the dimensionality. Yao et al. [37] introduce the kernel approximately harmonic projection (KAHP) which is a suitable strategy to reduce dimensionality of the data in combination with the *K*-means algorithm in a finite-dimensional context. Chen and Li [7] propose the use of a Johnson–Lindenstrauss type random projection as a preprocessing for functional learning algorithms. These ideas are adapted to this work in a context of functional data clustering, as Biau et al. [3]. A second strategy based on spectral methods is introduced.

We discuss numerical aspects of *K*-means algorithms for cluster analysis in infinite-dimensional Hilbert spaces. The proposed algorithm, named KK-means, consists of applying the *K*-means algorithm to functional data using projections on RKHS. The contributions of this paper can be summarized as follows:

- An interpretation of the two-stage methods based on principal component analysis is given. These approaches calculate the distance between the empirical functions as the distance between their projections onto function spaces. In this paper two types of projections have been proposed (in two different bases) using the Tikhonov regularization theory.
- A numerical study is conducted using (i) the *K*-means algorithm applied to a raw data, (ii) the approximate kernel *K*-means (aKKm, [8]) and (iii) the KK-means algorithm. By means of the Kappa coefficient of agreement, it is shown that if the data are sampled at regular time intervals the functional nature of the data can be ignored without damaging the performance of the clustering procedure. This is a numerical validation of the theoretical result of [5]. A numerical counter-example has been included, with a non-uniform discretization strategy, for raw-data clustering methods in which the performance of aKKm and the *K*-means algorithm is significantly deteriorated compared with the KK-means algorithm, which accounts for the functional nature.
- The effectiveness of a dimension reduction strategy based on the number of eigenfunctions has been numerically demonstrated. It was found that when data are uniformly sampled, KK-means and aKKm have a significantly lower computational cost than a *K*-means algorithm applied to raw data.

In Section 2 we develop metrics for empirical functions and establish relationships to calculate them by the Euclidean distance between projections in finite dimensional spaces. In Section 3 we carry out a numerical experiment to evaluate the performance of the proposed algorithms and analyse a case study. Finally, in Section 4 we gather together the conclusions obtained.

2. Projecting functional data onto reproducing Kernel Hilbert spaces

Fig. 1 shows the contents of this section schematically. As a first step, we project the empirical functions onto a Reproducing Kernel Hilbert Spaces (RKHS) $\mathcal{H}_K$ applying the Tikhonov regularization theory in RKHS. Following González-Hernández [15] we use two representations of these projections, one using kernel expansions and the other eigenfunctions. We define the distance between two empirical functions as the distance in $\mathcal{H}_K$ between their projections. Secondly, we use Cholesky decomposition to show that this distance between functions can be calculated by the Euclidean distance between a data transformation. This theoretical result presents two meaningful advantages: (i) it allows the use of the vectorial *K*-means algorithm codes for functional data and (ii) as the Johnson–Lindenstrauss-type random projections to reduce the dimensionality requires working with Euclidean distance, this transformation enables it to be used. Other theoretical contributions are set out in Appendix A which it shows that the proposed scheme (depicted in Fig. 1) coincides with functional principal component analysis (FPCA) in which projections are performed using the Tikhonov regularization theory. This identification allows us to interpret the FPCA as a cluster method in which the similarity measure is the distance between projected functions.

This section begins with a brief revision of RKHS and the Tikhonov regularization theory in RKHS. The general theory of

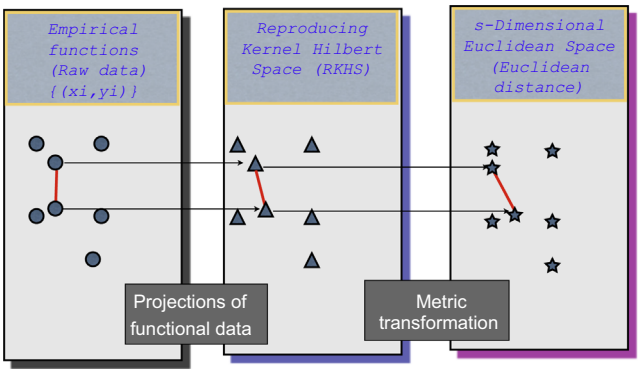

Fig. 1. Schematic representation of the clustering approach.

Tikhonov regularization is in the book of [34] and the general theory of RKHS is in [2].

Definition 1 (*Reproducing Kernel*). Let $\mathcal{H}$ be a real Hilbert space of functions defined in $X \subset \mathbb{R}^p$ with inner product $\langle \cdot,\cdot \rangle_{\mathcal{H}}$. A function $K : X \times X \mapsto \mathbb{R}$ is called a Reproducing Kernel of $\mathcal{H}$ if

1. $K(\cdot, x) \in \mathcal{H}$ for all $x \in X$.
2. $f(x) = \langle f, K(\cdot x) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$ and for all $x \in X$.

We define the norm by $\|f\|_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}^{1/2}$

A Hilbert space of functions that admits a Reproducing Kernel is called a *Reproducing Kernel Hilbert Space* (RKHS). The reproducing Kernel of a RKHS is uniquely determined. Conversely, if K is a positive definite and symmetric kernel (Mercer kernel), then it generates a unique RKHS in which the given kernel acts like a Reproducing Kernel.

For our purpose we list the fundamental properties of RKHS:

Theorem 1. *Let $\mathcal{H}$ be a real Hilbert space of functions defined in $X \subset \mathbb{R}^p$ with inner product $\langle \cdot,\cdot \rangle_{\mathcal{H}}$. Let $K : X \times X \mapsto \mathbb{R}$ be a Reproducing Kernel of $\mathcal{H}$. Then it follows that*

(a) $K(x,y) = \langle K(\cdot,x), K(\cdot,y) \rangle_{\mathcal{H}}$ *for all* $x, y \in X$.
(b) $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(x_i, x_j)$ *for all* $f \in \mathcal{H}$ *where*

$$f = \sum_{i=1}^{n} \alpha_i K(\cdot, x_i)$$

with $x_i \in X$.
(c) *Let* $L_K(f)(x) = \int_X K(x,y) f(y)\, \mathrm{d}y$ *with* $f \in L^2(X)$. *Mercer's Theorem asserts that*

$$K(x,y) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(y),$$

where ϕ_j *is the j-th eigenfunction of* L_K *and* λ_j *its corresponding non-negative eigenvalue.*
(d) *The function space* $\mathcal{H}$ *is given by*

$$\mathcal{H} = \left\{ f \in L^2(X) : \sum_{j=1}^{\infty} \lambda_j^{-1} \left[\left\langle f, \phi_j \right\rangle_{L^2(X)} \right]^2 < \infty \right\} \tag{1}$$

and the inner product can be written as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \lambda_j^{-1} \left\langle f, \phi_j \right\rangle_{L^2(X)} \left\langle g, \phi_j \right\rangle_{L^2(X)} \text{ with } f, g \in \mathcal{H}. \tag{2}$$

Now we briefly describe the Tikhonov discrete regularization for the problem at hand. Let K be a Mercer kernel and $\mathcal{H}_K$ its associated RKHS. Consider a subset compact $X \subset \mathbb{R}^p$ and let ν be a Borel probability measure in $X \times \mathbb{R}$. Let the regression function

$$f_\nu(x) = \int_{\mathbb{R}} d_\nu(y|x) \tag{3}$$

where $d_\nu(y|x)$ is the conditional probability measured on $\mathbb{R}$. Both ν and f_ν are unknown and what we want is to reconstruct this mean function.

Let

$$X_n := \{x_1, \ldots, x_n\} \subset X$$

and let f_n be a random sample independently drawn from ν and f_ν on X. That is

$$f_n := \{(x_i, y_i) \in X \times \mathbb{R}\}_{i=1}^n.$$

The Tikhonov regularization considers the function space

$$V_n := \mathtt{span}\{K(\cdot, x) : x \in X_n\} \tag{4}$$

where span is the linear hull and projects f_ν onto this space by using the sample f_n. The Tikhonov regularization theory makes a stable reconstruction of f_ν by solving the following optimization problem:

$$f^* := \arg\min_{f \in V_n} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \gamma \|f\|_{\mathcal{H}_K}^2 \tag{5}$$

where $\gamma > 0$ and $\|f\|_{\mathcal{H}_K}$ represents the norm of f in $\mathcal{H}_K$. The solution f^* of (5) is called the *Regularized γ-Projection* of f_ν onto $\mathcal{H}_K$ associated to the sample f_n.

The Regularized γ-Projection of f_n^* belongs to $V_n \subset \mathcal{H}_K$ and for this reason it can be expressed as a linear combination of any basis of $\mathcal{H}_K$. In the next subsection we calculate these projections onto two basis of $\mathcal{H}_K$. The first basis consists of the functions $\{K(x, x_i) : x_i \in X_n\}$ and the second one consists of *eigenfunctions* $\{\phi_1, \phi_2, \cdots\}$ of the integral operator L_K.

2.1. Kernel representation and associated distance

The *representation* theorem gives a closed form solution of f^* for the optimization problem (5). This theorem was introduced by Kimeldorf and Wahba [19] in a spline smoothing context and has been extended and generalized to the problem of minimizing risk of functions in RKHS, see [31,10].

Theorem 2 (*Representation*). *Let f_n be a sample of f_ν, let K be a (Mercer) kernel and let $\gamma > 0$. Then there is a unique solution f^* of (5) that admits a representation by*

$$f^*(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i), \quad \textit{forall } x \in X, \tag{6}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$ is a solution to the linear equation systems:

$$(\gamma n \mathbf{I}_n + K_{\mathbf{x}})\boldsymbol{\alpha} = y, \tag{7}$$

where $\mathbf{I}_n$ is the identity matrix $n \times n$, $y = (y_1, \ldots, y_n)^T$ and the matrix $K_{\mathbf{x}}$ is given by $(K_{\mathbf{x}})_{ij} = K(x_i, x_j)$. The expression (6) leads to the estimate of f_ν in X_n

$$\widehat{f}^* = K_{\mathbf{x}} \alpha \tag{8}$$

Theorem 1(b) establishes that the norm of f in V_n is given by the following inner product:

$$\|f\|_{V_n}^2 = \alpha^T K_{\mathbf{x}} \alpha, \quad \text{for all } f \in V_n. \tag{9}$$

As $K_{\mathbf{x}}$ is a symmetric positive definite matrix it admits the decomposition

$$K_{\mathbf{x}} = V^T D V \tag{10}$$

where the rows of the matrix V are the eigenvectors of the matrix $K_{\mathbf{x}}$ and D is a diagonal matrix whose diagonal entries are the corresponding (non-negative) eigenvalues. Therefore,

$$K_{\mathbf{x}} = (D^{1/2}V)^T (D^{1/2}V) = U^T U \tag{11}$$

with $U = D^{1/2}V$. Using the Cholesky decomposition (11)

$$\|f\|_{V_n}^2 = \alpha^T K_{\mathbf{x}} \alpha = \alpha^T U^T U \alpha = \tilde{\alpha}^T \tilde{\alpha} = \|\tilde{\alpha}\|^2 \tag{12}$$

with $\tilde{\alpha} = U\alpha$. This expression shows that the norm of $f(x) = \sum_{i=1}^n \alpha_i K(x, x_i)$, with $x_i \in X_n$, in V_n coincides with the Euclidean norm of a transformation of the space, that is $\|\tilde{\alpha}\|$.

Definition 2 (*Distance d_{V_n}*). Let X be a compact set and ν, μ two Borel probability measures defined on $X \times \mathbb{R}$. Let $K : X \times X \mapsto \mathbb{R}$ be a Mercer kernel and $\mathcal{H}_K$ its associated RKHS. Let f_ν and g_μ be functions defined as (3). Let $f_n := \{(x_i, y_i) \in X \times \mathbb{R}\}_{i=1}^n$ and $g_n := \{(x_i, y_i') \in X \times \mathbb{R}\}_{i=1}^n$ be two samples of the previous functions obtained from the probability distributions ν and μ. Suppose that the Regularized γ-Projection of these functions are $f^*(x) = \sum_{i=1}^n \alpha_i K(x, x_i)$ and $g^*(x) = \sum_{i=1}^n \beta_i K(x, x_i)$.

We define the square of the distance d_{V_n} from f_n to g_n by

$$d^2_{V_n}(f_n,g_n) := \|f^*-g^*\|^2_{V_n} = (\alpha-\beta)^T K_{\mathbf{x}}(\alpha-\beta)$$
$$= (\alpha-\beta)^T U^T U(\alpha-\beta) = \|U(\alpha-\beta)\|^2$$
$$= \|U\alpha - U\beta\|^2 = \|\tilde{\alpha}-\tilde{\beta}\|^2 \quad (13)$$

where $\tilde{\alpha} := U\alpha$ and $\tilde{\beta} := U\beta$.

2.2. RKHS representation and associated distance

González-Hernández [15] introduces an alternative representation for sampling functions in RKHS.

Theorem 3 (*RKHS representation*). *Let X be a compact set and let ν be a Borel probability measure defined on $X \times \mathbb{R}$. Let $K : X \times X \mapsto \mathbb{R}$ be a Mercer kernel and $\mathcal{H}_K$ its associated RKHS. Let f_ν be defined by (3) and $f_n = \{(x_i, y_i) \in X \times \mathbb{R}\}$ a sample of f_ν drawn from the probability distribution ν. Let L_K be the integral operator associated with the kernel K and let $\{\lambda_1, \lambda_2, \ldots\}$ be the eigenvalues of L_K and $\{\phi_1, \phi_2, \ldots\}$ the corresponding eigenfunctions. Then the projection f^* given by the minimization of (5) can be written by*

$$f^*(x) = \sum_j \alpha_j^\phi \phi_j(x) \quad (14)$$

where α_j^ϕ are the weights of the projection of f^* onto the function space RKHS generated by the eigenfunctions $\{\phi_1(x), \phi_2(x), \ldots\}$. In practice, when a finite sample is available, the first $s \le \operatorname{rank}(K_{\mathbf{x}})$ weights α_j^ϕ can be estimated by

$$\widehat{\alpha}_j^\phi = \frac{\ell_j}{\sqrt{n}}(\alpha^T v_j), \quad j = 1, \ldots, s; \quad (15)$$

where ℓ_j is the j-th eigenvalue of the matrix $K_{\mathbf{x}}$, v_j is the j-th eigenvector of $K_{\mathbf{x}}$ and α is the solution to Eq. (7). This leads to the approximation:

$$f^*(x) = \sum_j \alpha_j^\phi \phi_j(x) \cong \sum_{j=1}^{s} \widehat{\alpha}_j^\phi \phi_j(x). \quad (16)$$

The functions $\phi_j(x)$ are unknown but they can be approximated at the sampling points x_i by $\widehat{\phi}_j(x_i) = \sqrt{n} v_{ji}$ to obtain the following approximation:

$$\widehat{f}^{\,*} \cong \sqrt{n} \sum_{j=1}^{s} \widehat{\alpha}_j^\phi v_j, \quad (17)$$

where $\widehat{f}^{\,*} = (\widehat{f}^{\,*}(x_1), \ldots, \widehat{f}^{\,*}(x_n))^T$.

Definition 3 (*Empirical γ-regularized distance d_ϕ*). Let X be a compact set and let ν and μ be two Borel probability measures defined on $X \times \mathbb{R}$. Let $K : X \times X \mapsto \mathbb{R}$ be a Mercer kernel and $\mathcal{H}_K$ its associated RKHS. Let f_ν and g_μ be functions defined as (3). Let $f_n := \{(x_i, y_i) \in X \times \mathbb{R}\}_{i=1}^n$ and $g_n := \{(x_i, y_i') \in X \times \mathbb{R}\}_{i=1}^n$ be two samples of the previous functions drawn from the probability distributions ν and μ. Suppose that the Regularized γ-Projection of these functions is $f^*(x) \cong \sum_{j=1}^s \widehat{\alpha}_j^\phi \phi_j(x)$ and $g^*(x) \cong \sum_{j=1}^s \widehat{\beta}_j^\phi \phi_j(x)$.

We define the square of the empirical distance d_ϕ from f_n to g_n by

$$d^2_\phi(f_n,g_n) := \|f^*-g^*\|^2_{\mathcal{H}_K} = \langle f^*-g^*, f^*-g^* \rangle_{\mathcal{H}_K} \quad (18)$$

$$= \sum_{j=1}^{\infty} \lambda_j^{-1} \left\langle f^*-g^*, \phi_j \right\rangle_{L^2(X)} \left\langle \phi_j, f^*-g^* \right\rangle_{L^2(X)} \quad (19)$$

$$= \sum_{j=1}^{s} \frac{(\widehat{\alpha}_j^\phi - \widehat{\beta}_j^\phi)^2}{\lambda_j}. \quad (20)$$

Following González-Hernández [15] and Smale and Zhou [32] the eigenvalues and eigenvectors of $K_{\mathbf{x}}/n$ converge to the eigenvalues and eigenfunctions of L_K, and we have

$$\lambda_j \cong \frac{\ell_j}{n} \quad (21)$$

and we obtain the following approximation:

$$d_\phi(f_n,g_n) \cong \sqrt{\sum_{j=1}^{s} \frac{n(\widehat{\alpha}_j^\phi - \widehat{\beta}_j^\phi)^2}{\ell_j}} = \|\tilde{\alpha}^\phi - \tilde{\beta}^\phi\| \quad (22)$$

where $\|\cdot\|$ is the Euclidean norm, $\tilde{\alpha}^\phi := \sqrt{n} D_s^{-1/2} \widehat{\alpha}^\phi$, $\tilde{\beta}^\phi := \sqrt{n} D_s^{-1/2} \widehat{\beta}^\phi$ and $D_s^{-1/2}$ is the matrix $s \times s$

$$D_s^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{\ell_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\ell_2}} & \cdots & 0 \\ 0 & 0 & \cdots & \frac{1}{\sqrt{\ell_s}} \end{pmatrix}.$$

2.3. Dimensionality reduction strategies for each distance d_{V_n} and d_ϕ

The functional data, due to their nature, in most cases are of very high dimension. This leads the application of the cluster algorithms to require high CPU times and intensive use of the RAM. In this subsection we discuss (i) how to reduce the original data representation according to the distance d_ϕ or d_{V_n} and (ii) the relationship between the two distances.

First we discuss the question (ii). Let us see that the distances d_{V_n} and d_ϕ coincide for $s = n$. The distances d_{V_n} and d_ϕ project the original data and thus the Euclidean distance for projected data is calculated. Let us check that for $s = n$ both projections coincide. Let $\tilde{\alpha}^\phi$ and $\tilde{\alpha}$ be the projection of the function f_n onto RKHS and the kernel respectively. In this case $D = D_n$ and the coefficients $\widehat{\alpha}^\phi$, given by (15), are written in the matrix form:

$$\widehat{\alpha}^\phi = \frac{1}{\sqrt{n}} DV\alpha \quad (23)$$

therefore if $s = n$

$$\tilde{\alpha}^\phi = \sqrt{n} D_s^{-1/2} \widehat{\alpha}^\phi = D^{-1/2} DV\alpha = U\alpha = \tilde{\alpha}. \quad (24)$$

This is the expected result since the distances are based on two representations in different basis of the same function. The essential difference between these two approaches is how they reduce the dimension of the data.

2.3.1. Dimensionality reduction by working with d_ϕ

According to the relationships set out in Appendix A with FPCA, it follows that a dimension reduction strategy must be based on the magnitude of the eigenvalues $\{\ell_j\}$. Let $\{\ell_1, \ldots, \ell_s\}$ be the s largest eigenvalues of $K_{\mathbf{x}}$ and V_s the matrix whose rows are the s eigenvectors. In this case the matrix of projection is

$$P_s = D_s^{1/2} V_s. \quad (25)$$

2.3.2. Johnson–Lindenstrauss-type random projections to reduce dimensionality by working with d_{V_n}

We now analyze Johnson–Lindenstrauss-type random projections to reduce the dimensionality n of the data as described in [3]. The key idea that enables the use of these projections is that the distances between the empirical functions are calculated by the Euclidean distance of the transformed data.

Given s a positive integer and s sequences of independent random variables $(M_{1,i})_{i \ge 1} \cdots (M_{s,i})_{i \ge 1}$ with normal distribution with mean 0 and variance $1/s$, we define

$$M_s := \begin{pmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & \vdots & \vdots \\ M_{s1} & \cdots & M_{sn} \end{pmatrix}.$$

A random projection using the linear function $M:\mathbb{R}^n \mapsto \mathbb{R}^s$ with $M(\alpha)=M_s\cdot\alpha$ is defined. The next lemma [3] states how to choose the value of s.

Lemma 1 (*Johnson–Lindenstrauss Lemma*). *For any $\varepsilon,\delta\in(0,1)$ and any positive integer N, let s be a positive integer such that*

$$s\geq 4(\varepsilon^2/2-\varepsilon^3/3)^{-1}\log\frac{N}{\sqrt{\delta}}.$$

Then, for any set $\mathcal{D}$ of N points and for all $(\tilde{\alpha},\tilde{\beta})\in\mathcal{D}\times\mathcal{D}$ with probability $1-\delta$, the following holds:

$$(1-\varepsilon)\|\tilde{\alpha}-\tilde{\beta}\|\leq\|M_s\tilde{\alpha}-M_s\tilde{\beta}\|\leq(1+\varepsilon)\|\tilde{\alpha}-\tilde{\beta}\|. \tag{26}$$

The Johnson–Lindenstrauss lemma guarantees that the distance derived from the standard Euclidean product is preserved in the set of N original data points. Applying this to our case:

$$d_{V_n}(f_n,g_n)=\|\tilde{\alpha}-\tilde{\beta}\|\cong\|M_sU\alpha-M_sU\beta\| \tag{27}$$

where $\tilde{\alpha}:=U\alpha$ and $\tilde{\beta}:=U\beta$. The matrix of projection, for the metric d_{V_n}, is

$$P_s=M_sU. \tag{28}$$

2.4. The K-means algorithm for functional data

In this section we synthesize the above results into a K-means algorithm for functional data. In essence, the functional data are projected onto a s dimensional space and the K-means algorithm in finite-dimensional spaces is applied to the functional data. The K-means algorithm for functional data is described in Table 1.

Remark 1. This paper addresses the critical issue of cluster analysis in calculating distance between empirical functions so these results are efficiently applicable to many clustering schemes. Observe that once the functional data are projected in Step 1, any other clustering algorithm, such as hierarchical clustering, could have been applied in Step 2.

3. Numerical trials

In this section several numerical experiments are performed. Our main goals are:

- To determine when it is necessary to consider explicitly the functional nature of the data that we wish to analyse. To do this, we will compare numerically the basic K-means algorithm and a kernel K-means algorithm, specifically the approximate kernel K-means (aKKm) applied to raw data with the KK-means algorithm.
- To determine the advantages of considering distances derived from the functional projections.
- To analyze the effect of the dimensionality reduction parameter s used to d_ϕ in the performance of KK-means algorithm.
- To provide the methodology described with a real example and to identify what difficulties must be addressed.

The first three questions are addressed in *Experiments*1a–1c and the third goal in *Experiment*2.

3.1. Description of the numerical experiments

3.1.1. Test problems

The test problems used in the numerical experiments are:

- `Waves`: These data are obtained by a convex combination of triangular waves [4] defined as follows (see Fig. 2):

  $y_1(x)=uh_1(x)+(1-u)h_2(x)+\varepsilon(x)$ for the class 1,
  $y_2(x)=uh_1(x)+(1-u)h_3(x)+\varepsilon(x)$ for the class 2,
  $y_3(x)=uh_2(x)+(1-u)h_3(x)+\varepsilon(x)$ for the class 3,

  where u is a uniform random variable in $(0,1)$; $\varepsilon(x)$ is a standard normal variable and the functions h_k are the following triangular waves in $x\in[1,21]$:

  $h_1(x):=\max(6-|x-11|,0);\quad h_2(x)=h_1(x-4);\quad h_3(x)=h_1(x+4)$

  The data generated are shown in Fig. 3.
- `Signals` with different type of noise: This test problem is described in [21]. It starts with a piecewise linear function which is perturbed by three different types of noise. In this problem we aim to decide if K-means algorithm for functional data would be able to group signals according to their type of noise since all the signals share the same non-random trend.
- `Spectrometric`: This dataset is made up of the performance of the infrared absorbtion spectrum of meat samples. Each

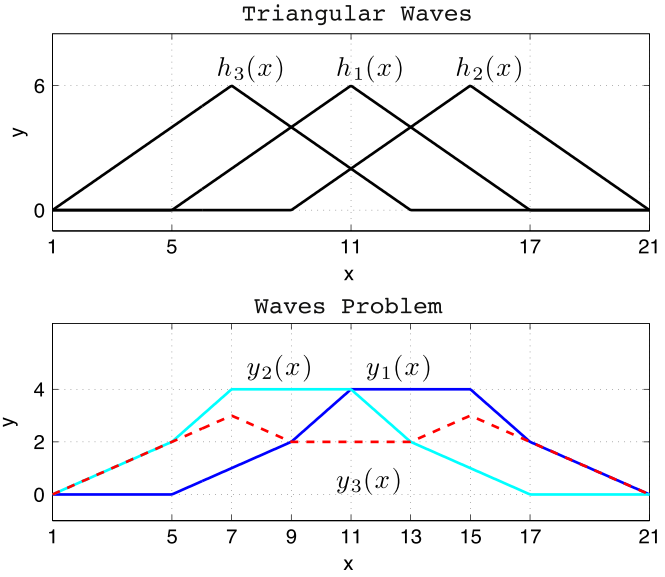

Fig. 2. Illustration of the `Wave` problem generation.

Table 1
The proposed K-means algorithm for functional data (KK-means).

Step 0. (*Initialization*). Given $X_n:=\{x_1,\ldots,x_n\}\subset X$ and a sample of empirical functions $\{f^j\}_{j=1}^N$ where $f^j:=\{(x_i,y_{ij})\in X\times\mathbb{R}: i=1,\ldots,n\}$, choose a type of kernel $K(\cdot,\cdot)$ and its parametrization. Calculate the kernel matrix K_x as $K_{\mathbf{x}}=K(x_i,x_j)$. Choose the regularization parameter γ. Solve the following N linear equation systems:
$(\gamma n\mathbf{I}_n+K_{\mathbf{x}})\boldsymbol{\alpha}=y^j,\quad j=,\ldots,N$ (29)
where $y^j=(y_{1j},\ldots,y_{nj})^T$. Denote by α^j those solutions

Step 1. (*Projection*). Choose a dimension s to represent data and one of the projections (25) or (28). Project the original data using the expression:
$\overline{\alpha}^j=P_s\alpha^j,\quad j=1,\ldots,N.$ (30)

Step 2. (*The K-means algorithm in $\mathbb{R}^s$*). Apply the K-means algorithm to the dataset $\{\overline{\alpha}^j\}_{j=1}^N$

Step 3. (*Calculation of the centroid*). Calculate the centroids in the original space V_n or in RKHS

observation consists of an absorbtion spectrum in 100 wavelengths that vary from 850 to 1050 nm. Furthermore, for each sample a chemical analysis has been carried out to determine the fat content. Each class is determined by those samples with less than 30% of fat content and more than 30%. These data are available at http://www.math.univ-toulouse.fr/staph/npfda.

- `Phonemes`: This database gathers the pronunciation of the 5 phonemes ("sh", "iy", "dcl", "aa", "ao") emitted by 400 individuals. These data are the *log-periodograms* discretized at 150 points and are available in http://www.math.univ-toulouse.fr/staph/npfda.
- `Irrationals`: We have constructed the following three irrational functions:

$$y_1(x) = x(1-x) + \varepsilon(x) \text{ for class 1,}$$

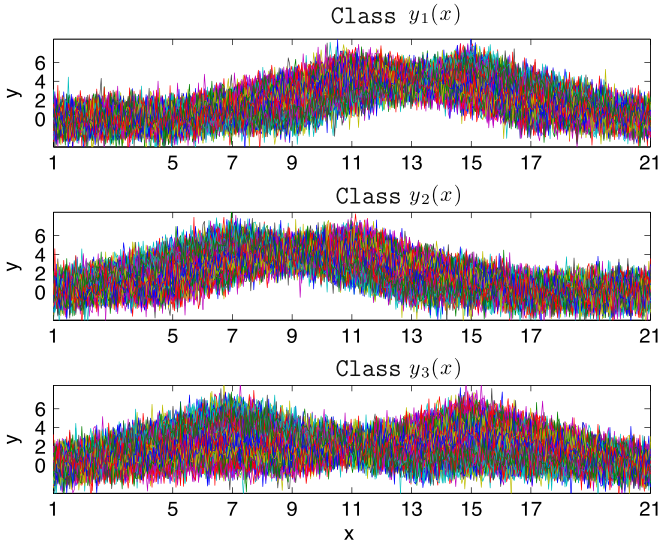

Fig. 3. `Waves` problem.

$$y_2(x) = x^{3/2}(1-x) + \varepsilon(x) \text{ for class 2,}$$

$$y_3(x) = x^2(1-x) + \varepsilon(x) \text{ for class 3,}$$

where $\varepsilon(x)$ is a uniform random variable in $(-0.2, 0.2)$. We have generated 100 functions for each class. In this example the sampling of the region X is not uniform. On one hand, the interval $(0.1, 0.9)$ has been uniformly sampled at 9 points and on the other hand $(0.9, 1]$ has been uniformly sampled at 1000 points. The generated data are shown in Fig. 4. The main characteristic of this example is that the sampling X is not uniform.

- `50Words`, `Adiac`, `MedicalImages`, `SwedishLeaf`, `Synthetic-control` and `WordsSynonyms`: In order to provide a comprehensive evaluation, we have added these six diverse time series datasets, from the UCR Time Series repository. The essential characteristic of those problems is that they have a large number of clusters. The dataset `Synthetic-control` is synthetic, i.e. created by some researchers to test some property. The datasets `50Words`, `MedicalImages` and `WordsSynonyms` are real, i.e. they were recorded as natural time series from some physical process. Finally, the datasets `Adiac` and `SwedishLeaf` are shape type, these are one-dimensional time series that were extracted by processing some two-dimensional shapes. All these time series were sampled regularly in time. The data are available in [18].
- `Radar`: We consider 472 radar signals obtained from the Topex/Poseidon satellite in a 25 mile band of the Amazon river. Each wave is associated with a type of terrain and these data are used in hydrology and altimetry. These data are real and the true number of clusters is unknown. The data are available in http://www.math.univ-toulouse.fr/staph/npfda/. *Experiment2* has been carried out on this dataset.

Table 2 shows the number of objects, the number of sample points and the number of original clusters.

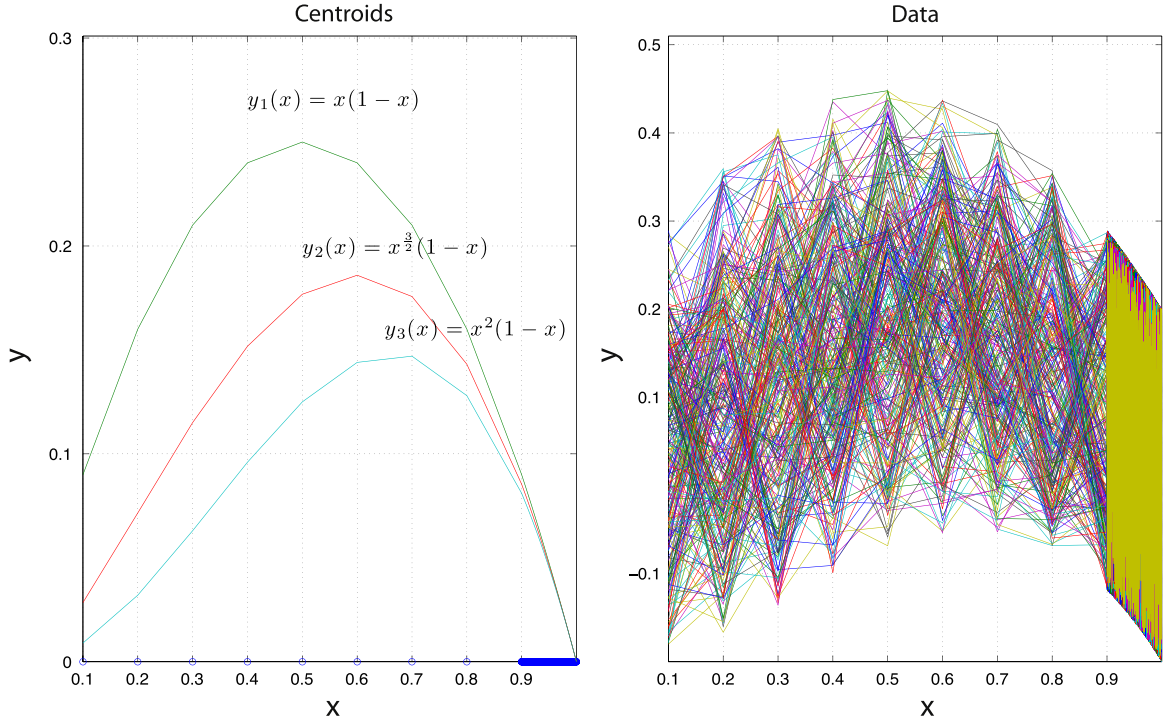

Fig. 4. `Irrationals` problem.

Table 2
Statistics of test problems.

Problem	Clusters (K)	Objects (N)	Sample size (n)
Waves	3	450	400
Signals	3*	150	100
Spectrometric	2	215	100
Phonemes	5	2000	150
Irrationals	3	300	1009
50Words	50	905	270
Adiac	37	781	176
MedicalImages	10	1141	99
SwedishLeaf	15	1125	128
Synthetic-control	6	600	60
WordsSynonyms	25	905	270
Radar	?	472	70

* There are three types of functions regarding the way the noise of the signals is generated, but a unique cluster if we consider the non-random piece of the function.

Table 3
Kernel parameters used in the numerical tests.

Problem	σ_1	σ_2	a	b
KK-means				
Waves	1	1	1.0e+00	5
Signals	10	10	1.0e−03	5
Spectrometric	1	1	1.0e−04	5
Fonemes	1	1	1.0e−04	5
Irrationals	10	5	1.0e−02	5
50words	1	1	1.0e−02	5
Adiac	1	1	1.0e−03	5
MedicalImages	1	1	1.0e−03	5
SwedishLeaf	10	1	1.0e−03	5
Synthetic-control	1	10	1.0e−03	5
WordsSynonyms	1	1	1.0e−03	5
aKKm				
Waves	40	36	1.0e−02	5
Signals	30	95	1.0e−04	5
Spectrometric	3	1	1.0e−04	5
Fonemes	15	23	2.0e+02	5
Irrationals	12	5	1.0e−01	5
50words	18	24	1.0e−03	5
Adiac	2	0.3	1.0e−02	5
MedicalImages	4	3	1.0e−03	5
SwedishLeaf	10	1	1.0e−03	5
Synthetic-control	7	9	1.0e−03	5
WordsSynonyms	13	10	1.0e−03	5

3.1.2. Proposed methods

In numerical experiments we consider the following algorithms:

1. K-means: Each functional datum is given by a sampling $\{(x_i, y_i)\}_{i=1}^n$. The K-means algorithm applied to this type of data ignores the values of x_i and uniquely considers the vector $(y_1, \ldots, y_n)$.
2. L-KK-means: We take the Laplacian kernel defined by

   $K(x_i, x_j) := e^{-1/\sigma_1^2 \|x_i - x_j\|_1}$

   where $\|\cdot\|_1$ is the norm 1 and $\sigma_1 \in \mathbb{R}$.
3. G-KK-means: We take the Gaussian kernel defined by

   $K(x_i, x_j) := e^{-1/\sigma_2^2 \|x_i - x_j\|_2^2}$

   where $\|\cdot\|_2$ is the Euclidean norm and $\sigma_2 \in \mathbb{R}$.
4. P-KK-means: In this case we take the polynomial kernel defined by

   $K(x_i, x_j) := (1 + a x_i^T \cdot x_j)^b$ with $a > 0$ and $b \in \mathbb{N}$.

5. Kernel K-means algorithms: We have considered approximate kernel K-means (aKKm) algorithm [8] as an important instance of this class. The desired goal is to compare these methods with those proposed in this work. In this case the kernel function is evaluated on the vector of images $\{y^i\}$, i.e. $K(y^i, y^j)$ and the Laplacian, Gaussian and polynomial kernels have also been considered .

The kernel parameters are shown in Table 3. The value of regularization parameter γ is 0.0001 for L-KK-means and G-KK-means methods. For P-KK-means method is $\gamma = 1$ in all the test except Irrationals problem where $\gamma = 0.0001$.

3.2. Experiment 1: numerical tests

3.2.1. Experiment 1a: performance of the K-means and KK-means algorithms

In this section the K-means and KK-means algorithms are going to be tested on the datasets. The results are evaluated by comparing the grouping obtained by the cluster algorithm with the original grouping. Two indices that are used by Xu et al. [36] to perform document clustering are the *accuracy* (AC) and the *index of the mutual information*. The first is defined by

$$AC = \frac{\sum_{i=1}^N \delta(\alpha_i, map(\beta_i))}{N}$$

where N is the number of functions and $\delta(x, y)$ is the delta function that is one if $x = y$ and 0 otherwise, and $map(\beta_i)$ that maps each label β_i with its equivalent in the original data. The Kuhn–Munkres algorithm (or Hungarian algorithm) has been used to find the best mapping $map(\beta_i)$ [23]. Note that this index matches the best observed proportion of agreement between the two groupings.

The Kappa Coefficient of Agreement κ [6] is a statistical measure of the degree of agreement between two *experts* and it could be an alternative to the use of the *mutual information index*. This index is also applied to cluster analysis. Each object of the sample is classified twice by the same category system. The goal is to evaluate if the observed agreement is higher than expected. This statistic is defined by

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{31}$$

where $P_o = \sum_{k=1}^K p(c_k, map(c_k))$ is the observed proportion of the objects in the sample (that is AC index) that have been classified in the same category by both experts (the cluster algorithm and the system of original categories) and $P_e = \sum_{k=1}^K p(c_k) p(map(c_k))$ is the expected proportion of agreement between categories. The numerator of κ is the proportion observed to be greater than expected and its denominator is the maximum value that the numerator could take. The range of possible values of κ is the interval $[-1, 1]$. This value is more suitable than AC because it corrects the effect of the expected agreement.

In *Experiment*1a we use the distance d_ϕ with the strategy of reduction of the dimensionality given in Section 2.3.2 where the number of eigenvalues s is indicated in Table 4.

The K-means algorithm works well for uniformly spaced functional data (see [5]). In our case, this is shown in Table 5 where the K-means algorithm gives κ values similar to those obtained with methods based on projections on RKHS in all the test problems with uniformly spaced data.[1] To research this matter further we consider the following distance in the function space $L^2[a, b]$ and an approximation using trapezoidal-rule of integration:

$$\|f(x) - g(x)\|_{L^2[a,b]}^2 = \int_a^b (f(x) - g(x))^2 \, dx \cong h\left(\frac{(f(a) - g(a))^2}{2} + \frac{(f(b) - g(b))^2}{2} + \sum_{i=1}^{n-1} (f(a + ih) - g(a + ih))^2\right)$$

[1] All the test problems, except Irrationals problem, the data $\{x_i\}_{i=1}^n$ are uniformly distributed on X.

where $h=(b-a)/n$. If we have a sample of $f(x)$ and $g(x)$ such that $f(a+ih)=f_i$ and $g(a+ih)=g_i$ and we denote f and g by the vectors $(f_1,\ldots,f_n)$ and $(g_1,\ldots,g_n)$, then

$$\|f(x)-g(x)\|^2_{L^2[a,b]} \cong h\|f-g\|^2 - h\left(\frac{(f(a)-g(a))^2}{2}+\frac{(f(b)-g(b))^2}{2}\right).$$

If the number of data n is large then we deduce the above expression, where $\|\cdot\|$ is the Euclidean norm, that

$$\|f(x)-g(x)\|^2_{L^2[a,b]} \cong h\|f-g\|^2 \propto \|f-g\|^2.$$

It shows that the K-means algorithm is equivalent to working, when the points are uniformly distributed, with a distance proportional to $\|f(x)-g(x)\|^2_{L^2[a,b]}$. This justifies the obtained results.

The most significant result is obtained in the `Irrationals` problem. This problem has been made *ex profeso* to show that if the points set $\{x_i\}$ does not sample uniformly the space X, then the K-means algorithm can reduce its performance.

Each algorithm is run 1000 times. For each sample we have calculated the average number of iterations to converge, the proportion of achieved successes, that is, the proportion of times that the algorithm stops at the best solution of the sample and CPU times to perform all the runs. The results are shown in Table 6 where it can be seen that all the algorithms exhibit similar behavior except for the execution time. Kernel projections based methods reduce significantly the dimensionality of the problems and contribute a meaningful saving in CPU time.

Table 4
Value of s used in test problems.

Problem	s	Problem	s
`Waves`	10	`Signals`	10
`Spectrometric`	30	`Phonemes`	10
`Irrationals`	10	`radar`	20
`50Words`	20	`Adiac`	20
`MedicalImages`	20	`SwedishLeaf`	20
`Synthetic-control`	20	`WordsSynonyms`	20

Table 5
Kappa coefficient of agreement κ.

Problem	K-means	L-KK-means	G-KK-means	P-KK-means
`Waves`	0.263	0.260	0.263	0.253
`Signals`	0.010	0.010	0.010	0.010
`Spectrometric`	0.448	0.309	0.308	0.326
`Phonemes.`	0.824	0.818	0.821	0.701
`Irrationals`	0.150	0.425	0.525	0.525
`50words`	0.349	0.380	0.372	0.197
`Adiac`	0.319	0.319	0.308	0.190
`MedicalImages`	0.198	0.193	0.173	0.260
`SwedishLeaf`	0.313	0.326	0.305	0.324
`Synthetic-control`	0.482	0.482	0.590	0.650
`WordsSynonyms`	0.248	0.245	0.242	0.206

3.2.2. Experiment 1b: the parameter s

The parameter s is essential for the performance of the KK-means algorithms. Small values of the parameter s reduce the computational cost, but too small values could lead to information being lost and computational performance being damaged. To analyze this issue KK-means algorithms have been run on test problems, analyzing the value of the concordance coefficient κ versus the parameter $s=1,\ldots,20$. The results are shown in Fig. 5. It is generally observed that the maximum performance of KK-means algorithms is achieved by small values of s showing that it is a viable strategy to reduce the dimensionality. Note that in `Waves` example the original classification arranges the data as y_1,y_2,y_3 whereas KK-means algorithms use the functions h_1,h_2,h_3 as category systems. This means that when the parameter s increases this discrepancy is enhanced.

3.2.3. Experiment 1c: perfomance of the approximate Kernel K-means

The essential difference between the kernel clustering methods proposed in the literature and the one developed in this paper is that they ignore the data $\{x_i\}$ (ignore functional nature) and work directly with the data points $\{y^i\}_{i=1}^N$ where $y^i\in\mathbb{R}^n$. Unlike d^2_ϕ and $d^2_{V_n}$, the kernel distance function is computed as

$$d^2_K(y^i,y^j) := K(y^i,y^i)+K(y^j,y^j)-2K(y^i,y^j) \tag{32}$$

where $K(\cdots,\cdot):\mathbb{R}^N\times\mathbb{R}^N\mapsto\mathbb{R}$ is a kernel function.

We have focused the study on the aKKm algorithm because this algorithm achieves better clustering performance than the traditional low rank kernel approximation and the running time and memory requirements are significantly lower than those of kernel K-means. Appendix A shows how the aKKm algorithm can be derived by applying the K-means algorithm on a transformation of the original data.

The aKKm algorithm samples $m\ll N$ points to approximate the kernel matrix $(K_{\mathbf{y}})_{ij}=K(y^i,y^j)$. The parameter m plays the role of data reduction, the same role as the parameter s in the KK-means algorithms. We have repeated Experiment 1a using the value $m=s$ and Laplacian, Gaussian and polynomial kernels. As the data are now $\{y^i\}_{i=1}^N$ instead of $\{x_i\}_{i=1}^n$ they have different orders of magnitude and it is necessary to calculate suitable parameters for these algorithms. The parameters used are shown in Table 3.

Table 6
Computational cost.

Problem	K-means			L-KK-means			G-KK-means			P-KK-means		
	Iter.	p	CPU (s)	Iter.	p	CPU (s)	Iter.	p	CPU (s)	Iter.	p	CPU (s)
`Waves`	7.8	0.94	112.3	7.8	0.96	12.7	8.3	0.95	14.1	14.1	0.11	19.1
`Signals`	4.7	0.26	29.9	9.2	0.23	8.8	9.7	0.62	12.3	12.9	0.07	29.9
`Spectrometric`	7.4	1.0	25.4	7.4	0.70	11.8	7.5	1.0	14.1	9.7	0.79	14.6
`Phonemes.`	20.6	0.21	1188.4	24.9	0.002	163.6	23.7	0.29	166.2	23.2	0.34	59.9
`Irrationals`	13.1	0.001	5047.9	15.8	0.029	26.5	13.1	0.365	12.5	12.4	0.188	12.6
`50words`	22.3	0.001	4461.3	22.5	0.001	334.5	22.2	0.001	330.9	24.5	0.001	245.8
`Adiac`	36.8	0.001	2746.3	35.6	0.001	364.0	36.4	0.001	371.7	42.4	0.001	306.1
`MedicalImages`	31.1	0.003	582.9	30.1	0.001	132.2	30.7	0.001	145.5	32.9	0.002	117.6
`SwedishLeaf`	32.5	0.001	1246.9	30.9	0.001	219.3	32.3	0.001	194.6	39.1	0.001	219.7
`Synthetic-control`	11.8	0.067	46.6	11.5	0.007	21.2	11.5	0.048	22.1	13.3	0.001	22.8
`WordsSynonyms`	26.7	0.001	3080.3	26.6	0.001	216.1	26.9	0.001	217.1	30.9	0.001	174.3

p: estimate of the probability the method will reach the global optimum. Iter.: the mean number of iterations.

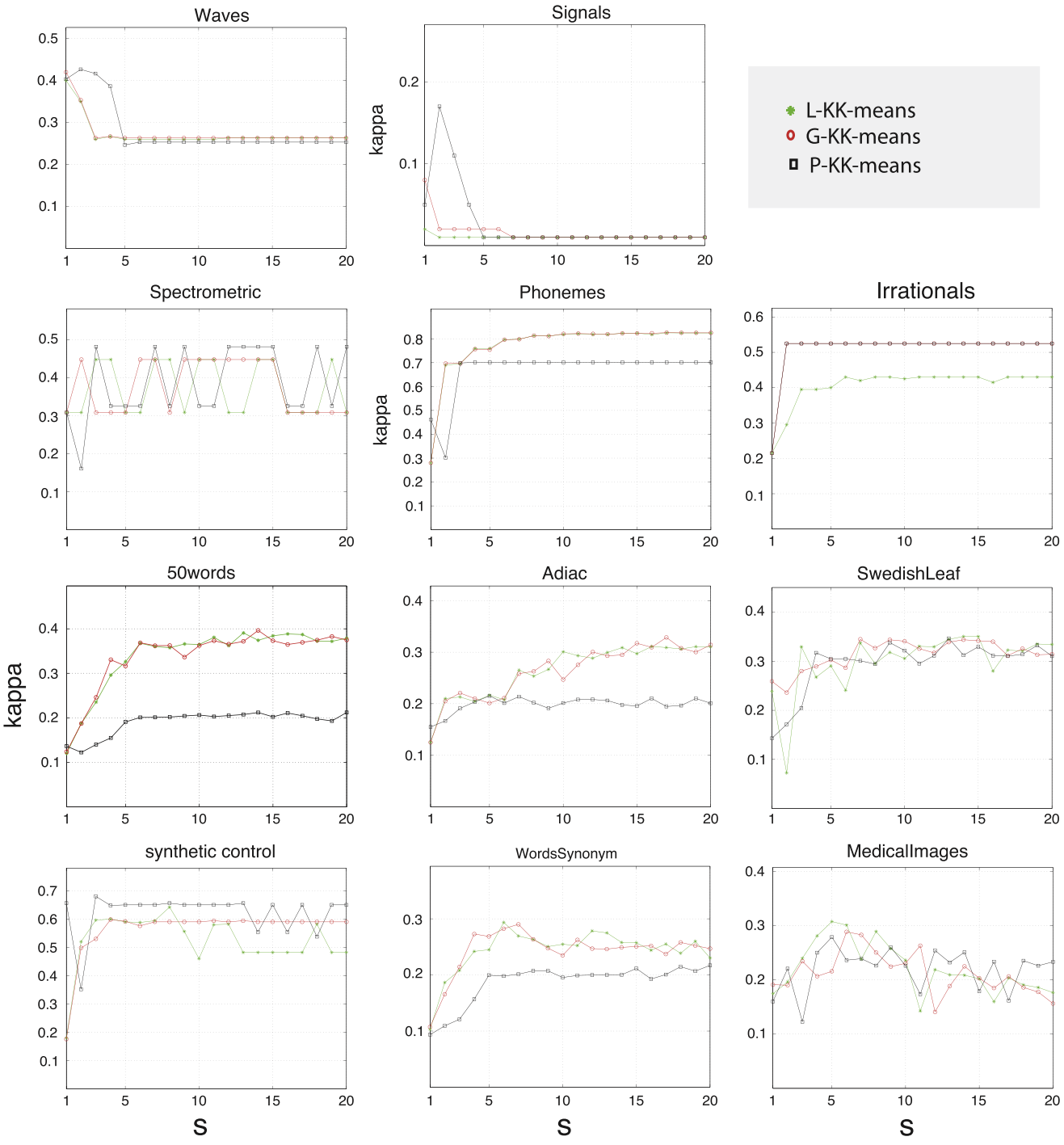

Fig. 5. Kappa coefficient versus the parameter s.

The results obtained are displayed in Tables 7 and 8. It can be observed that the performance of the algorithm is in general similar to KKmeans algorithms in almost all the problems. There are only two significantly different problems. The `irrationals` problem, from not considering the functional natureaKKm, has a degree of concordance of almost 0. On the other hand the `Signals` problem was not accurately addressed by the KK-means methods. This is because the type of noise defines the class in which each function is placed. Once an empirical function is projected, its noise is smoothed, and functions become indistinguishable from the KK-means algorithm. By contrast, the aKKm algorithm is able to identify the type of noise present in each signal.

3.3. Experiment 2: an application of functional cluster analysis for radar signals

3.3.1. Computation of the kernel and its parameters

In this section we apply the proposed methodology to the real `Radar` problem. The first task is to calculate the type of kernel, its parametrization and the regularization parameter γ. The expression (17) allows us to determine the projections and visually decide what type of kernel and values of γ are suitable. Fig. 6 shows the average across 472 collected signals in radar (red graphic) and the average across the corresponding projections obtained by Laplacian, Gaussian and polynomial kernels. These projections has been made for three different sets of parameters. With Polynomial kernel, we chose the

parameter $b=5$ in all the experiments. The γ took the values $10^0, 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4}. The role that plays this parameter is shown clearly in Gaussian kernel with $\sigma_2 = 10$. "Great" values of γ for the determined problem lead the projection to take the shape of the original curve, but the norm of the projection is much smaller than the norm of the original. Thus the parameter measurement γ involves taking $\gamma \to 0^+$ but without causing an ill-posed system (7). A polynomial kernel with $a=0.1$ and $\gamma = 10^{-4}$ exhibits an ill-posed system. A visual inspection allows the choice of $\sigma_1 = \sigma_2 = 10$ and $a=0.01$ and a regularization parameter $\gamma = 10^{-4}$. We also note that Gaussian and Laplacian kernels produce similar projections and get results reproducing the average signal.

3.3.2. Determining the number of clusters

A second problem we must address to apply K-means algorithm (for functional data or non-functional) is that the number of cluster K must be known. Halkidi et al. [16] classify the validity methods in external, internal and relative methods. The external and internal are statistical methods based on Monte Carlo simulation. The relative methods choose a defined *validity index* that is optimized with respect to a Clustering Parametrization, in this case with respect to the number of clusters K. In this numerical experiment we have chosen three indexes representing different contexts. A Xie-Beni index [35] has been applied to some fuzzy clustering algorithms, the coefficient of determination for hierarchical cluster analysis and F-Snedecor index for partitional methods such as the K-means algorithm. We define the following indexes:

1. Coefficient of determination R^2: This coefficient measures the proportion of variance explained by the model. It is defined by

$$R_K^2 = 1 - \frac{SS_{within}^K}{SS_{total}}, \quad K = 2, \ldots, K_{max} \tag{33}$$

where SS_{within}^K and SS_{total} are respectively the sum of squares in K-clusters and the total sum of squares, that is, SS_{within}^1. This statistic R_K^2 takes values in $(0, 1]$ and its value can be a guide to choosing the number of clusters.
Wilks' Lambda Statistic is usually applied to multivariate analysis of variance (MANOVA) and is associated with the coefficient of determination by the expression $R^2 = 1 -$ Wilks' Lambda. The Wilks' Lambda Statistic has been applied to cluster analysis to determine the number of clusters [20]. Kuo and Lin [20] determine the value that gives a larger decreased of the statistic by visual inspection, in our case, a larger increased of R^2 (that is, a non-smooth point that looks like a sharp point).
2. F-Snedecor index: Analysis of covariance (ANCOVA) is a parametric statistical method that tries to find a significant difference between the averages of certain groups but discounting the effects of the covariances (quantitative factors). Suppose we want to contrast the goodness of a statistical between Model 1 and another more complex model (with more parameters) that we call Model 2. The F statistic can be written by

$$F_{k_2-k_1, N-k_2} = \frac{(R_2^2 - R_1^2)/(k_2 - k_1)}{(1 - R_2^2)/(N - k_2)}, \tag{34}$$

where k_1 and k_2 are the number of parameters estimated for each of the models, R_1^2 and R_2^2 are the coefficient of determination obtained with the models 1 and 2 respectively and N is the number of observations. If we applied the above statistic to compare a partition with $k_1 = 1$ clusters to another with $k_2 = K$, the statistic would take the expression

$$\text{IndexF} = \frac{R_K^2/(K-1)}{(1 - R_K^2)/(N-K)}. \tag{35}$$

Observe that $R_1^2 = 1 - (SS_{within}^1/SS_{total}) = 1 - (SS_{total}/SS_{total}) = 0$.
The motivation for using this index is found in the work of Milligan and Cooper [27] which carried out an intensive research, based on Monte Carlo simulation analysis, for determining the correct number of clusters. They recommend maximizing the index

$$\left\{\frac{tr(B)}{K-1}\right\} \Big/ \left\{\frac{tr(W)}{N-K}\right\}. \tag{36}$$

The B and W terms are the between and pooled within cluster sum of squares and cross product matrices. Observe if we divide the numerator and denominator of the index (36) by the total sum of squares SS_{total} we obtain the index (35).
3. Xie-Beni index. It is defined by

$$I_{XB} := \frac{\sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk} d(f^j, v^k)}{N \min_{k \neq i} d(v^k, v^i)} \tag{37}$$

Table 7
Kappa coefficient of agreement κ for aKKm.

Problem	Laplacian aKKm	Gaussian aKKm	Polynomial aKKm
Waves	0.247	0.263	0.340
Signals	0.490	0.730	0.020
Spectrometric	0.154	0.328	0.448
Fonemes	0.781	0.839	0.466
Irrationals	0.010	0.010	0.010
50words	0.355	0.351	0.355
Adiac	0.246	0.359	0.329
MedicalImages	0.291	0.156	0.200
SwedishLeaf	0.503	0.270	0.315
Synthetic-control	0.450	0.582	0.452
WordsSynonyms	0.241	0.242	0.237

Table 8
Computational cost.

Problem	Laplacian aKKm			Gaussian aKKm			Polynomial aKKm		
	Iter.	p	CPU (s)	Iter.	p	CPU (s)	Iter.	p	CPU (s)
Waves	9.4	0.506	14.9	8.8	0.321	9.4	9.5	0.005	12.5
Signals	6.7	0.025	7.3	7.4	0.030	8.4	4.5	1.000	4.7
Spectrometric	7.6	0.012	9.5	6.9	0.363	6.2	6.2	0.757	5.4
Fonemes	17.1	0.464	41.4	16.3	0.160	35.5	21.4	0.107	51.0
Irrationals	7.5	0.002	13.6	17.5	0.001	33.5	5.5	0.394	9.8
50words	24.4	0.001	316.8	22.7	0.001	307.9	23.7	0.001	326.1
Adiac	35.0	0.001	246.5	24.5	0.001	179.0	32.1	0.001	279.2
MedicalImages	28.6	0.004	120.7	24.2	0.004	84.9	31.7	0.003	128.2
SwedishLeaf	29.0	0.001	151.6	25.6	0.001	120.3	33.4	0.001	208.3
Synthetic-control	11.0	0.004	22.6	9.7	0.058	21.6	11.6	0.002	25.8
WordsSynonym	28.4	0.001	211.6	27.1	0.001	201.5	26.6	0.001	198.3

p: estimate of the probability the method will reach the global optimum. Iter.: the mean number of iterations.

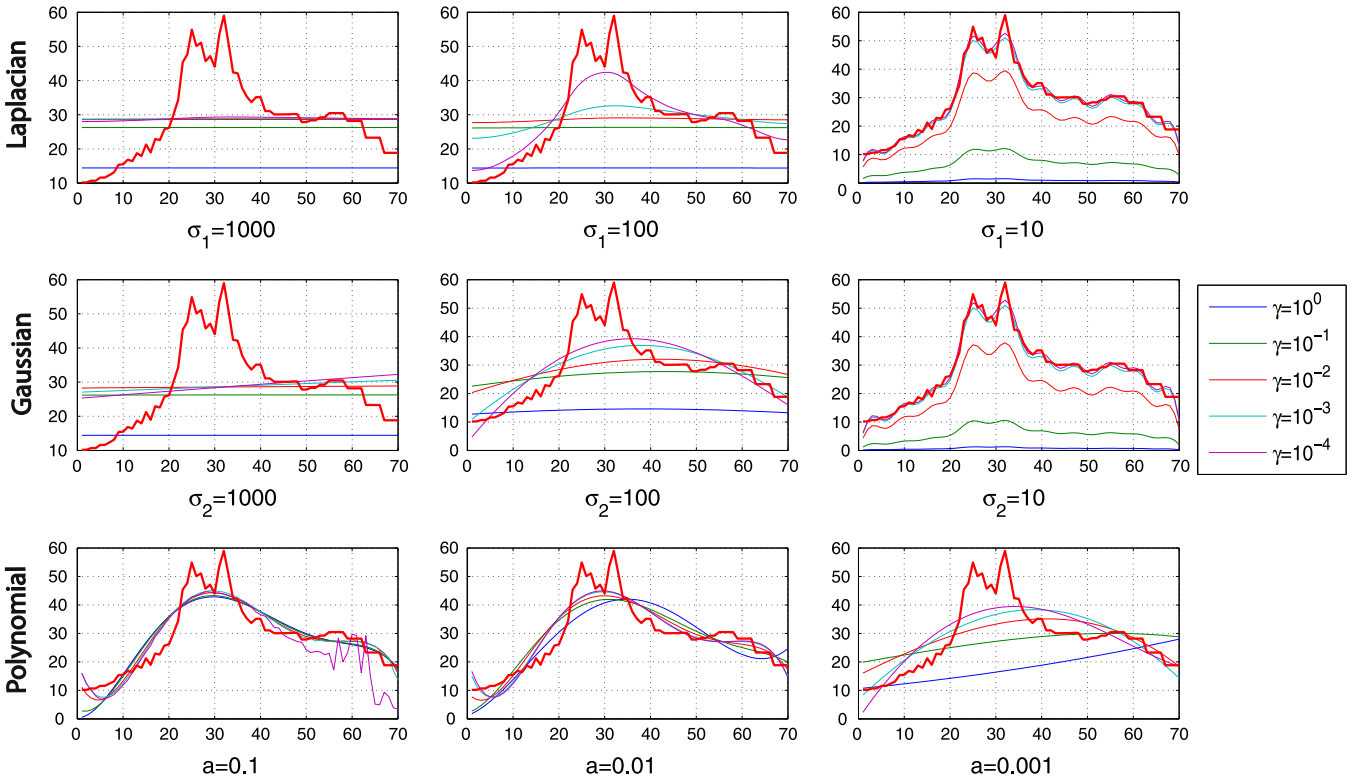

Fig. 6. Average projection for different kernels and parameters of the `Radar` problem. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

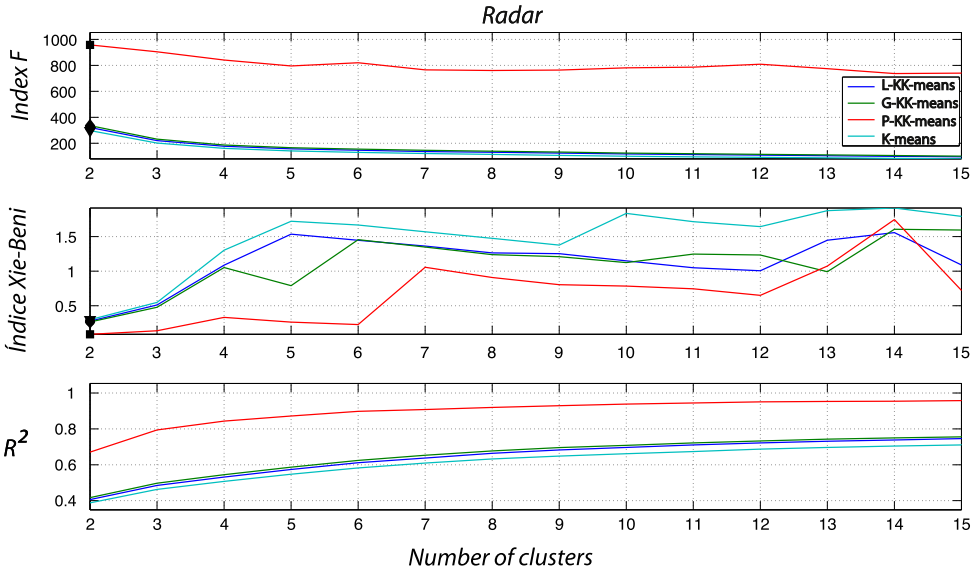

Fig. 7. Number of clusters in the `Radar` problem.

where $u_{jk}=1$ if the element j belongs to cluster k and takes the value 0 otherwise, v^i are the centroid of the clusters and f^j the original data.

Fig. 7 shows the coefficient of determination R^2 and the F and Xie-Beni indexes. The first observation is that K-means, L-KK-means and G-KK-means algorithms have a similar behavior, different from the behavior of the P-KK-means algorithm. The F and Xie-Beni indexes would take 2 clusters but there is little difference between 2 and 3 clusters. If we look at the coefficient of determination R^2 we find that $K=3$ has a sharp point (non-smooth). This criterion is used to determine the number of clusters, the one that produces an abrupt change (that leads to the appearance of a sharp point). In addition, the value of R^2 can be used as a selection criterion because it can be explained as the square of the proportion of variance. For 2 and 3 clusters we obtain $R^2 \approx 0.4$ and $R^2 \approx 0.5$, respectively. Finally taking into account the above considerations we choose $K^*=3$.

Fig. 8 shows the solution found by G-KK-means for $K^*=3$ clusters. The cluster 1 contains 94 signals, the cluster 2 has 47 signals and the third cluster has 329. The projected curves are in the left column and the original curves in the right. Fig. 9 shows the centroids (on the left the average of the projections and on the right the average of the original signals).

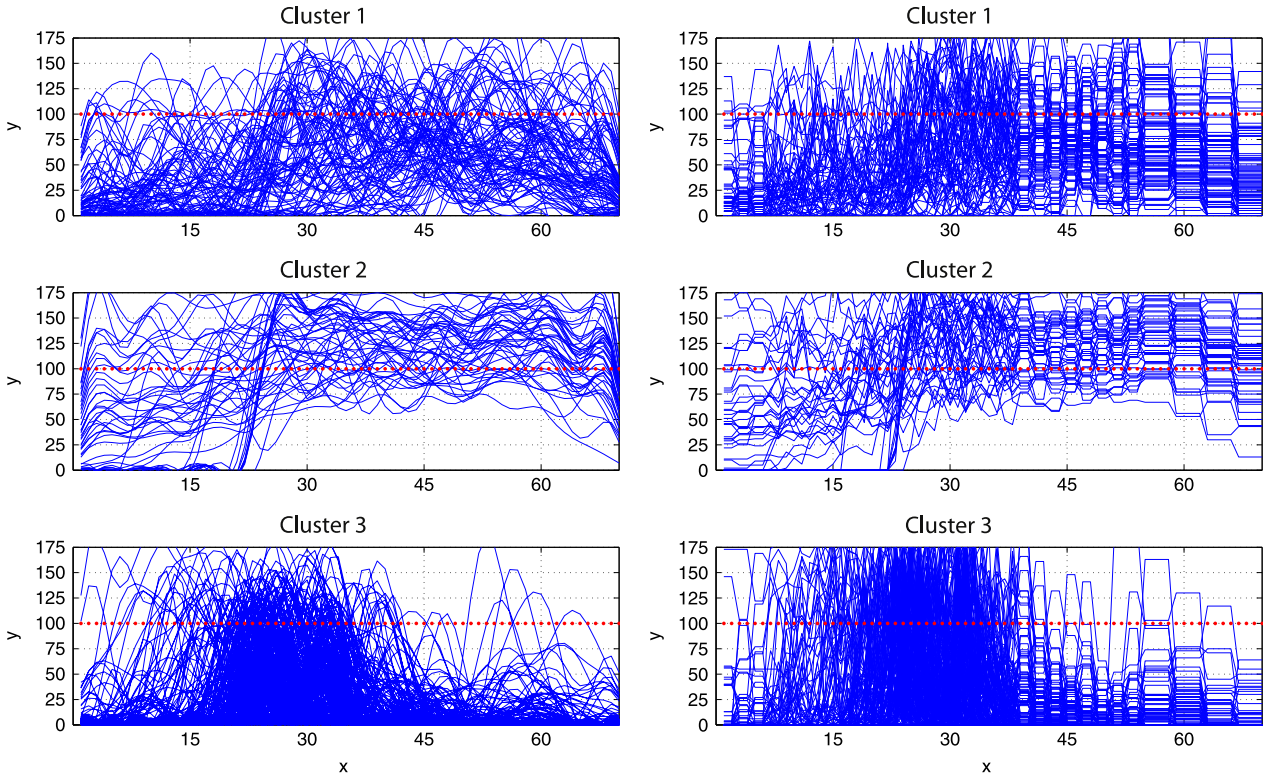

Fig. 8. Solution found in the `Radar` problem with G-KK-means. The curves in the left column are the projection and in the right column are the original curves.

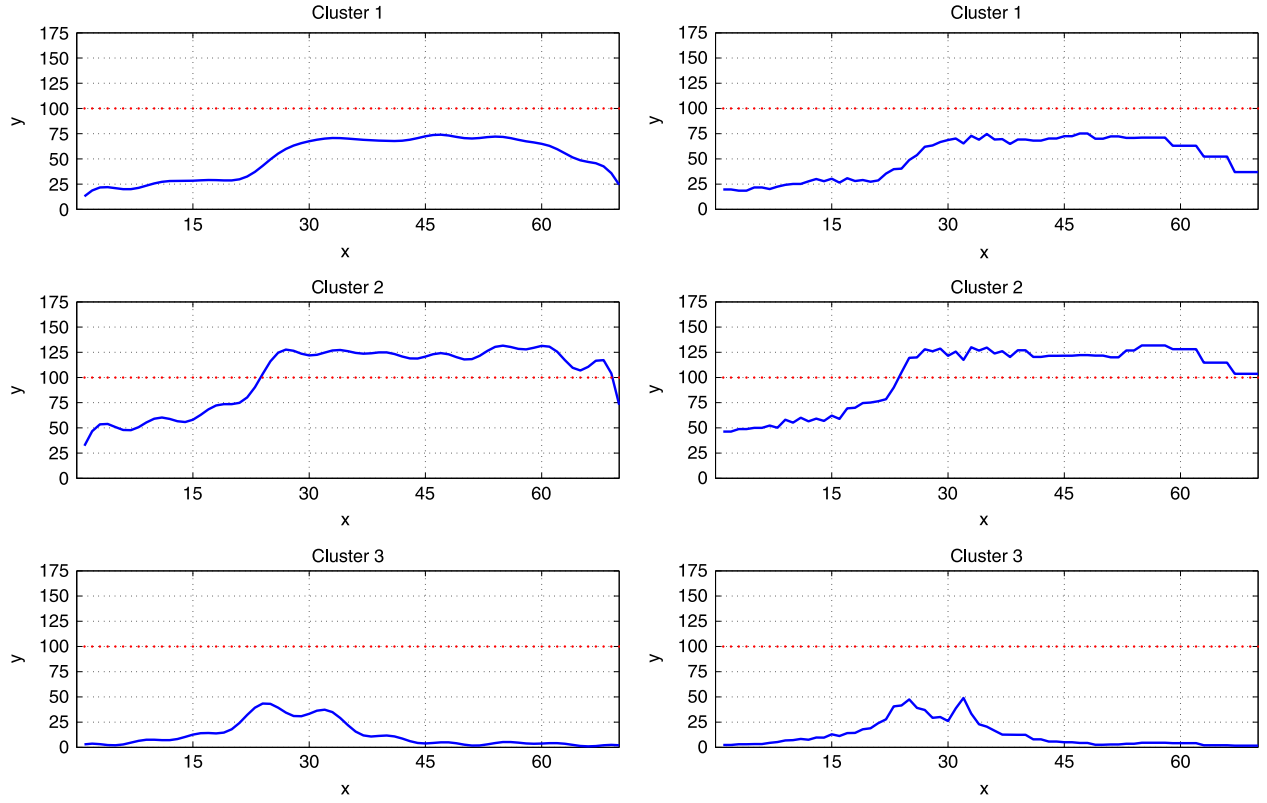

Fig. 9. Centroids of the solution found with G-KK-means in the `Radar` problem. The figure on the left shows the projection and on the right the original curves.

4. Conclusions

In this paper we have addressed the problem of cluster analysis for functional data. The proposed algorithm of *K*-means carried out by this work, named KK-means, solves two important challenges: (i) it is applicable to function domains where it is not possible to control the discretization $\{x_i\}_{i=1}^{n}$ of the functions and (ii) these discretizations lead to high dimensional problems due to the functional nature of the data and need strategies to reduce their dimensionality, in such a way that the performance of the procedure cluster is not reduced.

The metrics d_{V_n} and d_ϕ based on projections onto Reproducing Kernel Hilbert spaces (RKHS) and the Tikhonov regularization theory have been used. We have shown that both metrics are equivalent, however they lead, due to their nature, to two different strategies for reducing the dimension of the problem. In the case of d_{V_n} the most suitable strategies are Johnson–Lindenstrauss-type random projections. The dimensionality reduction for d_ϕ is based on spectral methods.

In the numerical study we have given an example *ex professo* to show that if the sampling $\{x_i\}_{i=1}^n$ is not uniform in X a K-means algorithm that ignores the functional nature of the data can reduce its performance. We have show numerically that for examples where X is sampled uniformly the performance of the original K-means algorithm and the aKKm algorithm is similar to the one proposed but the computational cost, when working with the set of original data, is larger than the KK-means algorithm based on d_ϕ or alternatively aKKm algorithm, because both algorithms use dimensionality reduction strategies.

It has been numerically proven that small values of the parameter s, less than 20, allow for maximum performance of the algorithms.

We have illustrated this methodology with a real problem to analyse what problems must be faced by the KK-means algorithm. For real functions of a single real variable the expression (17) allows a graphical representation that guides the choice of the kernel type, its parameters and the regularization parameter γ.

Determining the correct number of clusters is a hard task. Relying on a unique index can be misleading. R^2 is a good tool as it measures the proportion of variance and therefore choosing the minimum number of clusters to reach a specific value of R^2 can be a good criterion. The good thing about methods based on projections is that they eliminate data noise and allow a better estimation of R^2.

Acknowledgements

The authors wish to acknowledge financial support from *Ministerio de Economía y Competitividad* under Project TRA2011-27791-C03-03.

The authors express their gratitude to the anonymous referees, the associate editor and the editor whose comments greatly improved this paper.

Appendix A. Relationships between functional principal component analysis and d_{V_n} and d_ϕ

FPCA is a tool which represents curves in a function space of reduced dimension. In this appendix we show the relationship between FPCA and the projections used in this paper. In this appendix, we follow the description of FPCA for time series (which is more restrictive than the case discussed in the paper) by Jacques and Preda [17]. We assume that for a time series $f(t)$ the L^2-continuous stochastic process holds

$$\forall t \in [0,T], \quad \lim_{h\to 0} \mathbb{E}[(f(t+h)-f(t))^2] = 0. \tag{38}$$

Let $\mu(t) = \mathbb{E}[f(t)]$ be the mean function and the covariance operator $\mathcal{V}$ of f:

$$\mathcal{V}: L^2([0,T]) \mapsto L^2([0,T])$$
$$g \to \mathcal{V}(g) := \int_0^T V(\cdot,t)g(t)\,\mathrm{dt}$$

is an integral operator with kernel V defined by

$$V(s,t) = \mathbb{E}[(f(s)-\mu(s))(f(t)-\mu(t))], \quad s,t \in [0,T]. \tag{39}$$

The spectral analysis of $\mathcal{V}$ provides a countable set of positive eigenvalues $\{\lambda_j\}_{j\geq 1}$ associated with an orthonormal basis of eigenfunctions $\{f_j\}_{j\geq 1}$. The *principal components* $\{C_j\}_{j\geq 1}$ of $f(t)$ are random variables defined as the projection of f on the eigenfunctions of $\mathcal{V}$:

$$C_j = \int_0^T (f(t)-\mu(t))f_j(t)\,\mathrm{dt}. \tag{40}$$

The Karhunen–Loeve expansion holds

$$f(t) = \mu(t) + \sum_{j\geq 1} C_j f_j(t), \quad t \in [0,T]. \tag{41}$$

Truncating (41) at the first s terms one obtains the best approximation in norm L^2 of $f(t)$ by

$$f^{(s)}(t) = \mu(t) + \sum_{j=1}^{s} C_j f_j(t), \quad t \in [0,T]. \tag{42}$$

The computational methods for FPCA assume that the functional data belong to a finite dimensional space expanded by some basis of functions. Let $\alpha_i = (\alpha_{i1},\ldots,\alpha_{iL})^T$ be the expansion coefficient of the observed curve $\widehat{f}_i$ in the basis $\Phi = \{\varphi_1,\ldots,\varphi_L\}$ such that

$$\widehat{f}_i(t) = \Phi(t)^T\alpha_i \tag{43}$$

with $\Phi(t) = (\varphi_1(t),\ldots,\varphi_L(t))^T$.

Let $\tilde{A}$ be the $N\times L$-matrix whose rows are the vectors α_i^T, let $A = (I_N - \mathbb{1}_N(1/N,\ldots,1/N))\tilde{A}$ where I_N and $\mathbb{1}_N$ are respectively the identity $N\times N$-matrix and the unit column vector of size N and $W = \int_0^T \Phi(t)\Phi(t)^T\,\mathrm{dt}$ is the symmetric $L\times L$ matrix of the inner products between the basis functions. The functional principal component analysis is reduced to the usual PCA of the matrix $AW^{1/2}$.

The eigenfunction f_j belongs to the linear space spanned by the basis Φ:

$$f_j(t) = \Phi(t)^T b_j \tag{44}$$

with $b_j = (b_{j1},\ldots,b_{jL})^T$

The principal component scores are given by

$$C_j = AWb_j. \tag{45}$$

We now compare the FPCA with the KK-means algorithms. Observe that the principal components have zero mean whereas the projections $f^*(x)$ onto RKHS do not assume this condition. Operationally this means that working with the coefficient matrix A in the FPCA and with $\tilde{A}$ in the case of the paper. If we translate the data then d_{V_n} and d_ϕ are preserved. The use of A instead of $\tilde{A}$ is equivalent to the translation of data to the origin and both approaches are equivalent.

To interpret d_{V_n} under the FPCA, consider the basis $\varphi_j(t) := K(t,x_j)$ with $x_j \in X_n$. In this case, Theorem 1(a) guarantees

$$W = K_x = U^TU \Rightarrow W^{1/2} = U. \tag{46}$$

By using FPCA, the first principal components of $AW^{1/2} = AU$ are calculated. The distance d_{V_n} uses the transformation of the data AU as equivalent to considering all the principal components and working with the basis $\{K(\cdot,x_j)\}_{x_j\in X_n}$.

To interpret d_ϕ, we assume that the analytical expression for the kernel V is K and consider $\varphi_j = \phi_j$ the elements of the basis. Since $\{f_j\}$ is an orthonormal basis for $L^2([0,T])$ we have

$$f_j = \sqrt{\lambda_j}\phi_j \Rightarrow b_j = (0,\ldots,\sqrt{\lambda_j},\ldots,0)^T \tag{47}$$

and therefore the principal component is calculated

$$C_j = AWb_j = A\begin{pmatrix} \frac{1}{\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda_2} & \cdots & 0 \\ 0 & 0 & \cdots & \frac{1}{\lambda_s} \end{pmatrix} b_j \cong \sqrt{n}A\begin{pmatrix} \frac{1}{\sqrt{\ell_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\ell_2}} & \cdots & 0 \\ 0 & 0 & \cdots & \frac{1}{\sqrt{\ell_s}} \end{pmatrix} e_j$$
$$= \sqrt{n}AD_s^{-1/2}e_j \tag{48}$$

where e_j is a vector whose components are all zero except for the j-th component which is equal to one. This proves that the d_ϕ considers s principal components.

There are two novel aspects with respect to the classical FPCA. The first issue is that the expansion (43) is performed using the Tikhonov regularization theory. The second one is to interpret Euclidean distance between feature vectors (the principal components) as the distance between the projected functions.

Appendix B. Approximate kernel *K*-means (aKKm)

The purpose of this appendix is to show how the aKKm algorithm can be obtained by applying the algorithm *K*-means to a transformation of the original data. The main idea of these methods is the so-called *kernel trick*, which allows inner products to be computed in some, possibly infinite-dimensional, *feature space*. These methods are based on a nonlinear mapping $\Psi(\cdot)$ which projects the data representation in the original space onto the *feature space* $\mathcal{H}$. A *Mercer kernel* $K(\cdot,\cdot): Y \times Y \mapsto \mathbb{R}$ allows us to evaluate an inner product in the *feature space* by the expression:

$$K(y^i, y^j) = \left\langle \Psi(y^i), \Psi(y^j) \right\rangle_{\mathcal{H}} \tag{49}$$

where $\langle \cdot,\cdot \rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$.

The aKKm randomly samples m data points $m \ll N$, denoted by $\widehat{Y} = \{\hat{y}^1, \ldots, \hat{y}^m\}$ and construct a subspace $\mathcal{H}_b = \mathrm{span}[\Psi(\hat{y}^1), \ldots, \Psi(\hat{y}^m)]$. The kernel distance in the space $\mathcal{H}_b$ is computed by

$$d^2_{\mathcal{H}_b}(y^k, y^s) := \left\langle c_k(\cdot) - c_s(\cdot), c_k(\cdot) - c_s(\cdot) \right\rangle_{\mathcal{H}_b} = (\alpha_k - \alpha_s)^T \widehat{K} (\alpha_k - \alpha_s) \tag{50}$$

where

$$c_k(\cdot) = \sum_{i=1}^{m} \alpha_{ki} K(\hat{y}^i, \cdot);\ c_s(\cdot) = \sum_{i=1}^{m} \alpha_{si} K(\hat{y}^i, \cdot) \quad \text{and} \quad \widehat{K}_{ij} = K(\hat{y}^i, \hat{y}^j) \tag{51}$$

Assuming the matrix $\widehat{K}$ is well-conditioned[2] and using the Cholesky decomposition

$$\widehat{K} = U^T U.$$

The expression (50) can be calculated with Euclidean inner product

$$d^2_{\mathcal{H}_b}(y^k, y^s) = \left\langle U\alpha_k - U\alpha_s, U\alpha_k - U\alpha_s \right\rangle_2 \tag{52}$$

Chitta et al. [8] show that the relationship between the original data and the projections is given by

$$\alpha_k = \widehat{K}^{-1} \varphi_k^T \tag{53}$$

where φ_k is the e k-th row of the matrix $K_B \in \mathbb{R}^{N \times m}$ which is defined by $K_B(i,j) = K(y^i, \hat{y}^j)$.

Finally the above clustering method is equivalent to applying the *K*-means algorithm (Euclidean distance) to the transformed data (in rows)

$$K_B \widehat{K}^{-1} U^T = K_B U^{-1} (U^T)^{-1} U^T = K_B U^{-1}. \tag{54}$$

References

[1] C. Abraham, P.A. Cornillon, E. Matzner-Løber, N. Molinari, Unsupervised curve clustering using B-splines, Scand. J. Stat. 30 (3) (2003) 581–595.
[2] N. Aroszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (3) (1950) 337–404.
[3] G. Biau, L. Devroye, G. Lugosi, On the performance of clustering in Hilbert spaces, IEEE Trans. Inf. Theory 54 (2) (2008) 781–790.
[4] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
[5] B. Cadre, Q. Paris, On Hölder fields clustering, Test 21 (2) (2012) 301–316.
[6] J. Carletta, Assessing agreement on classification tasks: the kappa statistic, Comput. Linguist. 22 (1996) 249–254.
[7] D.-R. Chen, H. Li, On the performance of regularized regression learning in Hilbert space, Neurocomputing 93 (2012) 41–47.
[8] R. Chitta, R. Jin, T.C. Havens, A.K. Jain, Approximate kernel K-means: solution to large scale kernel clustering, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA, 2011, pp. 895–903.
[9] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[10] D. Cox, F. O'Sullivan, Asymptotic analysis of penalized likelihood and related estimators, Ann. Stat. 18 (1990) 1676–1695.
[11] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis, Springer Series in Statistics, Springer, New York, 2006.
[12] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (1) (2008) 176–190.
[13] C. Fyfe, P. Lai, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10 (5) (2001) 365–377.
[14] M. Girolami, Mercer kernel-based clustering in feature space, IEEE Trans. Neural Netw. 13 (3) (2002) 780–784.
[15] J. González-Hernández, Representing Functional Data in Reproducing Kernel Hilbert Spaces with Applications to Clustering, Classification and Time Series Problems (Ph.D. thesis), Department of Statistics, Universidad Carlos III, Getafe, Madrid, 2010.
[16] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2-3) (2001) 107–145.
[17] J. Jacques, C. Preda, Functional Data Clustering: A Survey, Research Report, No. 8198, INRIA, 2013.
[18] E. Keogh, Q. Zhu, B. Hu, Y. Hao. X. Xi, L. Wei, C.A. Ratanamahatana, The UCR Time Series Classification/Clustering Homepage: www.cs.ucr.edu/ eamonn/ time_series_data/, 2011.
[19] G. Kimeldorf, G. Wahba, A correspondence between Bayesian estimation on stochastic processes and smoothing splines, Ann. Math. Stat. 41 (2) (1970) 495–502.
[20] R. Kuo, L. Lin, Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering, Decis. Support Syst. 49 (4) (2010) 451–462.
[21] J. Łęski, A. Owczarek, A time-domain-constrained fuzzy clustering method and its application to signal analysis, Fuzzy Sets Syst. 155 (2005) 165–190.
[22] T.W. Liao, Clustering of time series data- a survey, Pattern Recognit. 38 (2005) 1857–1874.
[23] L. Lovász, M.D. Plummer, Matching Theory, vol. 367, American Mathematical Society, Chelsea Publishing, Providence, 2009.
[24] J. MacQueen, Some methods of classification and analysis of multivariate observations, in: L.M. Le Cam, J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, Berkeley, CA, 1967, pp. 281–297.
[25] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, Philos. Trans. R. Soc. Lond. Ser. A 209 (1909) 415–446.
[26] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.-R. Müller, Fisher discriminant analysis with kernels. in: Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing, Madison, USA, 1999, pp. 41–48.
[27] G. Milligan, M. Cooper, An examination of procedures for determining the numbers of clusters in a data set, Psychometrika 50 (1985) 159–179.
[28] J. Peng, H.-G. Müller, Distance-based clustering of sparsely observed stochastic, processes with applications to online auctions, Ann. Appl. Stat. 2 (3) (2008) 1056–1077.
[29] J.O. Ramsay, B. Silverman, Functional Data Analysis, Springer, New York, 2006.
[30] B. Schökopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[31] B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: Proceedings of the Annual Conference on Computational Learning Theory, 2001, pp. 416–426.
[32] S. Smale, D.X. Zhou, Geometry on Probability Spaces, Working Paper, 2007.
[33] D. Steinley, K-means clustering: a half-century synthesis, Br. J. Math. Stat. Psychol. 59 (1) (2006) 1–34.
[34] A. Tikhonov, V.Y. Arsenin, Solutions of Ill-Posed Problems, Wiley, New York, 1997.
[35] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[36] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the SIGIR, 2003, pp. 267–273.
[37] G. Yao, W. Hua, B. Lin, D. Cai, Kernel approximately harmonic projection, Neurocomputing 74 (17) (2011) 2861–2866.

[2] If the matrix $\widehat{K}$ had ill-conditioned problems, it could be replaced with $\gamma \mathbf{I}_m + \widehat{K}$ to a value $\gamma \in \mathbb{R}^+$ small enough.

María Luz López García is an Assistant Professor of Applied Mathematics at the University of Castilla-La Mancha. She received her M.Sc. degree in Mathematics from the *Universidad Autónoma de Madrid*, Spain, in 1990 and a Ph.D. from the *Universidad de Castilla-La Mancha*, Spain, in 2013. Her current research interests include data mining applied to transportation systems and artificial intelligence.

Ricardo García-Ródenas is an Assistant Professor of Applied Mathematics at the University of Castilla-La Mancha. He received a Mathematics degree from the *Universidad de Valencia*, Spain, in 1991 and a Ph.D. from the *Universidad Politécnica de Madrid*, Spain, in 2001. His current research interests include operational research in transportation systems and artificial intelligence.

Antonia González Gómez is an Assistant Professor of Applied Mathematics at the *University Politécnica de Madrid*. She received a Mathematics degree from the *Universidad Autónoma de Madrid*, Spain, in 1988 and a Ph.D. from the *Universidad Complutense de Madrid*, Spain, in 2003. Her current research interests include dynamic systems and artificial intelligence.