Time series classification methodology using reproducing kernel Hilbert spaces embedding

Student: Cristhian Kaori Valencia Marin

Supervisor: Andrés Marino Álvarez Meza, PhD

Universidad Tecnológica de Pereira
Engineering Faculty
Master in Electrical Engineering
Research Group in Automática
Pereira, Risaralda, Colombia
2019

Content

List of Figures

List of Tables

1 Symbols and Abbreviations

1.1 Symbols

It will be taken by notation, in bold capital letters the variables associated with matrices. Similarly, the vectors will be written in small letters and bold.

Symbol	**Definition**
d	distance
ζ	Kernel function
κ	Characteristic kernel function
ϕ	Mapping function
$\mathcal{X}$	input space
N	Number of samples
$\mathcal{H}$	Reproducing Kernel Hilbert Space
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Dot product in RKHS
σ	Characteristic kernel hyperparameter
μ	Marginal embedding operator
P, Q	Probability distribution functions
p, q	Probability density functions
$\hat{p}, \hat{q}$	Probability density functions estimation
α, β	Parameters probability density functions estimation
$\boldsymbol{\alpha}$	Parameters vector
$\mathbf{A}$	Matrix $\mathbf{A}$
$\mathbf{b}$	Vector $\mathbf{b}$
c	Scalar c
$\|c\|$	Absolute value of scalar c

1.2 Abbreviations

- **RKHS**:Reproducing Kernel Hilbert Space
- **HMMs**: Hidden Marcov Models
- **KAFs**: Kernel Adaptive Filters
- **QKLMS**: Quantized Kernal Least Mean Square
- **EM**: Espectation.Maximization

- **IPV**: Information Potential Variability
- **CKA**: Centered Kernel Alingment
- **MB**: Models-based
- **DB**: Distances-based
- **DTW**: Dynamic Time Warping
- **GPs**: Gaussian Processes
- **SVMs**: Support Vectors Machines
- **ED**: Euclidean distance
- **KL**: Kullback-Leibler
- **HSD**: HMMs stationary distance

2 Aknowledgments

I would like to thank to my supervisor Andrés M. Álvarez for an excellent guiding, for the advices and the motivation during this process. I also want to thank to Mauricio A. Álvarez and Edgar A. Valencia for introducing me into this exciting area of research and for insightful meetings. And I want to thank to the research group in Automática.

This work was funding for the project *Desarrollo de un sistema de apoyo al diagnóstico no invasivo de pacientes con epilepsia fármacoresistente asociada a displasias corticales cerebrales: método costo-efectivo basado en procesamiento de imágenes de resonancia magnética*; financed by Colciencias with code 1110-744-55778.

3 Abstract

Time series classification is a fundamental task in the areas of machine learning and pattern recognition, due to the multiple applications that exist in state of the art, such as analysis in stock markets, medicine, sensor networks, scientific experiments of moving objects, biology [1] and classification of forms [2]. Most data-based models assume that the observations are independent and identically distributed [3], [4]. However, by assuming the above, certain discriminating factors may be overlooked. Therefore, there is a need to represent the time series from models that take into account the sequential nature of the data. In the literature, there are two main approaches to the representation of time series: representation based on models (*Models-based, MB*) and representation based on distances (*Distances-based*, DB) [1]. However most of the MB depend on free parameters that must be previously tuned [5], [6], [7]. On the other hand, the DB approaches allow the construction of dissimilarity matrices in order to train classifiers based on close neighbors [1]. In this case, the most promising methods are based on RKHS embedding, because they allow to represent time series as points in Hilbert spaces of high dimensionality. However, these methods alone can not encode the information related to the temporal dependency, and in addition, the mapping to the RKHS depends strongly on the tuning of the parameter associated with the characteristic kernel [8]. This project seeks to build a methodology for the representation and classification of time series in RKHSs taking into account the temporal dependence and the automatic selection of the characteristic kernel hyperparameter.

4 Problem statement

In the real world, all physical phenomena can be understood as a time serie, that is why in the last decades the classification of time series has been gaining relevance [9], [10]. Sequences of numerical measurements occur at regular or irregular time intervals in vast quantities in almost all application domains, such as stock markets, medicine, sensor networks, scientific experiments on moving objects and biology [1]. Even the contours of static objects can be transformed into time series [11], [2], to which then the time series classification methods [12] can be applied.

Most data-based models assume that the observations are independent and identically distributed [3], [4]. However, by assuming the above, certain discriminating factors may be overlooked. Therefore, there is a need to represent the time series from models that take into account the sequential nature of the data. In the literature, there are two main approaches to the representation of time series: representation based on models (*Models-based*, MB) and representation based on distances (*Distances-based*, DD) [1]. In the first case, the MB representations allow coding the temporal dependencies of the data coming from time series from a set of parameters associated with the model, which take into account the relevance of each of the samples. Some of the most used models in the state of the art are: the hidden Markov models (HMMs) [5], the adaptive filters (AFs) [6], the Gaussian processes (GPs) [7], among others. HMMs allow to represent the data from a sequence of hidden states that encode the temporality of the samples; nevertheless, an appropriate choice of the topology of the model is required, that is, the form of the covariance matrix and the associated parameter to the number of hidden states [13]. In the case of AFs, they allow recursive learning of the time series using an optimal performance criterion and give prominence to the most relevant samples of the data set. However, for this type of models, the parameters associated with the quantization size and the error tolerance of the filter [14] must be properly tuned. The GPs are a stochastic process that allows Bayesian representation of a time series, taking into account the uncertainty of the data at each instant of time. Although GPs are nonparametric models, training these models is often computationally expensive due to the calculation of the inverse of a matrix that depends on the number of samples when calculating the posterior distribution [7]. On the other hand, the representation DD is based on the construction of a space of dissimilarity from matrices of distances between the data, which are later used to train classifiers based on close neighbors [15]. Within the main representations in spaces of dissimilarity are those based on the Euclidean distance, the distance Dynamic time warping (DTW) and the methodologies based on reproducing kernel Hilbert space (RKHS) embedding. Although the representations based on ED are the simplest and easiest to implement, it is necessary to clarify that this distance does not take into account the temporal dependence of the data [16], also, the ED can only be applied at the time of discriminate series of the same length [17]. The DTW is one of the most common algorithms in the state of the art when classifying time series because it can be seen as a

generalization of the ED exclusively for time series. In the case of the DTW, it is necessary to tune the parameter associated with the percentage of warping [18]. RKHS based methods are a promising approach used in various machine learning tasks nowadays. The basic idea of this method consists of the representation of probability distributions as points in an RKHS through an injective operator that depends on a characteristic kernel. This method has the following properties: it is a generalization of traditional methods based on kernel [19] and is very flexible when applied to high-dimensional statistical models [8]. However, these methods usually do not take into account the temporal dependence of the data, in addition to the classification performance that can be affected by the inadequate selection of the Hilbert space to which the time series is mapped, that is, it is necessary to find the optimal space in terms of separability between classes. This space depends closely on the bandwidth parameter of the characteristic kernel. In the state of the art this parameter is tuned in a heuristic way [20], [21], [22]. Based on the above, the following research question is posed: Is it possible to construct a methodology for the classification of time series from the mapping to RKHSs of sequential models that encode the temporary information and additionally build a scheme of the appropriate search of said space that guarantees the separability among classes?

5 Justification

Time series classification is a fundamental task in the areas of machine learning and pattern recognition, due to the multiple applications that exist in state of the art, such as analysis in stock markets, medicine, sensor networks, scientific experiments of moving objects, biology [1] and classification of forms [2].

One of the approaches in the state of the art are the representations from models that are suitable for modeling time series since they encode the temporal dependence of the data. Some of the most used models are HMMs, AFs, and GPs. However, most of these models depend on free parameters that must be previously tuned [5], [6], [7]. On the other hand, there are distance-based approaches that allow the construction of dissimilarity matrices in order to train classifiers based on close neighbors [1]. In this case, the most promising methods are based on RKHSE, because they allow representing time series as points in Hilbert spaces of high dimensionality. Without a system, these methods alone can not encode the information related to the temporal dependency, and in addition, the mapping to the RKHS depends strongly on the tuning of the parameter associated with the characteristic kernel [8].

With the development of this work, we seek to develop a methodology that allows the construction of a hybrid representation, that is, based on models and distances, that allows the sequential models to be mapped to the RKHS in order to codify the temporal dependence as well as to build classifiers based on dissimilarity. It also seeks to implement strategies for the adequate search of the RKHS that guarantees better results when classifying the time series.

6 Objectives

6.1 General Objective

To develop a methodology for time series classification using reproducing kernel Hilbert spaces, that allows finding the Hilbert space with the most suitable reproductive kernel in order to improve the separability between classes, regardless of the model used to learn the time series.

6.2 Specific objectives

1. To develop a measure of distance in reproducing kernel Hilbert spaces between sequential models learned from time series.
2. To implement a reproducing kernel Hilbert space selection method, which allows finding the highest separability between samples using unsupervised learning.
3. To implement a reproducing kernel Hilbert space selection method, which allows finding the highest separability between samples using supervised learning.

7 Background

In this section we describe the fuoundations required to introduce the proposed methodology as well as some previous time series classificaton definitions which motivates this work. Firtsly, we describe the fundamentals of RKHS embeddings and the construction of the RKHS-based metric how a dissimilarity measure in order two discriminate between two general models learned from time series, represented as probability distributions. Finally, describe some models used to estimate the RKHS-based metric.

7.1 Fundamentals of RKHS embeddings

7.1.1 Kernel-based methods

Kernel-based methods are based on the theoretical framework of Hilbert spaces generated by kernels (Mercer's Theorem), allowing non-linear versions of linear algorithms [23]. These methods are based on the kernel function (or kernel), $\kappa(x, y) : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$, which can be seen as a special case of a measure of similarity between two observations of the same input space $\mathcal{X}$. Formally a kernel $\kappa(\cdot, \cdot)$ is a dot product $\langle \cdot, \cdot \rangle$ in a hight dimensional space, possibly infinite, called features space $\mathcal{H}$. Let $\mathcal{X}$ a observations set, and $x, y \in \mathcal{X}$, then

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}, \tag{1}$$

where $\phi : \mathcal{X} \to \mathcal{H}$ is a non-linear mapping of the input space $\mathcal{X}$ to features space $\mathcal{H}$.

Another concept that will be used in this work is that of the kernel matrix or Gram matrix. Let a samples set $i.i.d$ $\{x_l\}_{l=1}^{N}$, the kernel matrix is denoted as $\boldsymbol{\kappa}$ and it's inputs are defined as $\kappa_{i,j} = \kappa(x_i, x_j)$. If $\boldsymbol{\kappa}$ is used to evaluate the dot products in a feature space $\mathcal{H}$ with mapping function ϕ, then the inputs associated with the kernel matrix are given by

$$\kappa_{i,j} = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}},$$

where $i, j = 1, 2, \ldots N$.

7.1.2 Reproducing kernel Hilbert Space

Let a topological space $\mathbb{R}$, and a kernel function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$, then a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel $\kappa(x, x')$, for $x, x' \in \mathcal{X}$, is a space of functions $g : \mathcal{X} \to \mathbb{R}$ that fulfill the following properties:

1. For all $x \in \mathcal{X}, \kappa(x,\cdot) : \mathcal{X} \to \mathcal{R}$, where $\kappa(x,\cdot) \in \mathcal{H}$.
2. $\langle g(\cdot), \kappa(x,\cdot)\rangle_H = g(x)$.

From these properties the following properties arise as a consequence

a) $\langle \kappa(x,\cdot), \kappa(y,\cdot)\rangle_H = \kappa(x,y)$.

b) $\kappa(x,x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$.

The proof can be seen in [24].

7.1.3 Distance between probability distributions on Hilbert space embeddings

Let $\mathcal{P}$ be the space of all probability distributions and let $X, Y \subset \mathcal{X}$ be two random variables that follow the distribution functions P and Q, respectively; then P, $Q \in \mathcal{P}$, $x \in X$, and $y \in Y$. Let $\mu(\cdot)$ be a marginal embedding operator that maps samples $x \in X$ to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, as follows:

$$\mu(P) = \mathbb{E}_X[\phi(x)] = \int_{\mathcal{X}} \phi(x) dP(x),$$

where $\mu : \mathcal{X} \to \mathcal{R}$ and $\phi : \mathcal{X} \to \mathcal{H}$. This embedding of probability distributions into RKHS allows us to compute distances between them. According to [25], the RKHS-based distance $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ over the probability measures P and Q, yields:

$$d^2_{\text{HE}}(P,Q) = \|\mu(P) - \mu(Q)\|^2_{\mathcal{H}}\,. \tag{2}$$

Equation (2) can be rewritten as

$$d^2_{\text{HE}}(P,Q) = \left\| \int_{\mathcal{X}} \phi(x) dP(x) - \int_{\mathcal{X}} \phi(y) dQ(y) \right\|^2_{\mathcal{H}}. \tag{3}$$

Afterward, we define a function $\kappa(x,x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$, $\forall x, x' \in \mathcal{X}$ as a reproducing characteristic kernel on $\mathcal{H}$ [26]. If the probability distributions $P(x)$ and $Q(y)$ admit density functions $p(x)$ and $q(y)$, respectively, we have $dP(x) = p(x)dx$ and $dQ(y) = q(y)dy$, then equation (3) can be written as

$$\begin{aligned} d^2_{\text{HE}}(P,Q) &= \int_{\mathcal{X}}\int_{\mathcal{X}} \kappa(x,x')p(x)p(x')dxdx' + \int_{\mathcal{X}}\int_{\mathcal{X}} \kappa(y,y')q(y)q(y')dydy' \\ &- 2\int_{\mathcal{X}}\int_{\mathcal{X}} \kappa(x,y)p(x)q(y)dxdy. \end{aligned} \tag{4}$$

The expression (4), is an analytical metric function in RKHS for probability distributions, that is, d_{HE} must fulfill the following properties [26]:

- $d_{\text{HE}}(x, y) \geq 0$.
- $d_{\text{HE}}(x, y) = 0 \rightarrow x = y$.
- $d_{\text{HE}}(x, y) = d_{\text{HE}}(y, x)$.
- $d_{\text{HE}}(x, z) \leq d_{\text{HE}}(x, y) + d_{\text{HE}}(y, z)$.

In this paper we seek to construct RKHS-based metrics, then we train a 1-NN classifier whose distance among neighbors is the RKHS metric. Note that distance equation (4) depends to the probability density functions (PDFs), $p(x)$ and $q(y)$, and the characteristic kernel κ. Therefore, both PDFs and κ should be previously estimated.

7.2 Probability density function estimation

In order to code the relevance on each of the samples in the metric (4), we built estimators for PDFs. There are two approaches for PDFs estimation: parametric and nonparametric models. In the case of no-parametric models, there is Gaussian Mixture Models (GMMs) and its extended version based on Parzen windows [6]. Analytically, the density functions $p(x)$ and $q(y)$ can be computed as a dot product between a function $f \in \mathcal{F}$ and the mapping φ into the RKHS $\mathcal{F}$, e.g,

$$p(x) = \langle f, \varphi(x) \rangle_{\mathcal{F}} = \int_{\mathcal{X}} \alpha_{x'} \varphi(x) dx', \tag{5}$$

where $\varphi : \mathcal{X} \rightarrow \mathcal{F}$ and $\alpha_{x'} \in [0, 1]$. Next, we rewrite $d^2_{\kappa}(P, Q)$ in terms of the α and β parameters as follows

$$\begin{aligned} d^2_{\text{HE}}(P, Q|\alpha_x, \beta_y) &= \int_{\mathcal{X}} \int_{\mathcal{X}} \alpha_x \alpha_{x'} \kappa(x, x') \zeta(x, x') dx dx' + \int_{\mathcal{X}} \int_{\mathcal{X}} \beta_y \beta_{y'} \kappa(y, y') \zeta(y, y') dy dy' \\ &- 2 \int_{\mathcal{X}} \int_{\mathcal{X}} \alpha_x \beta_y \kappa(x, y) \zeta(x, y) dx dy, \end{aligned} \tag{6}$$

where $\zeta(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$. The expression in (6) can be seen as a weighted enhancement of the well-known maximum mean discrepancy distance (MMD) with regard to the mapping function φ in the $p(x)$ and $q(y)$ estimation. Note that in the equation (6), the expression

$$\int_{\mathcal{X}} p(x) dx = \int_{\mathcal{X}} \int_{\mathcal{X}} \alpha_{x'} \zeta(x, x') dx' dx = 1,$$

then, the kernel ζ should be normalized, that is, $\int_{\mathcal{X}} \zeta(x, x') dx' = 1$. Thus, the sum of all weights $\alpha_{x'}$ is the unity,

$$\int_{\mathcal{X}} p(x) dx = \int_{\mathcal{X}} \alpha_{x'} dx = 1.$$

In general, the density of probability estimation can be computed as:

$$\hat{p}(x) = \sum_{x' \in \mathcal{X}} \alpha_{x'} \zeta(x, x') \qquad \hat{q}(y) = \sum_{y' \in \mathcal{X}} \beta_{y'} \zeta(y, y'). \tag{7}$$

Hence, the estimation for distance function in the expression (6) as given as

$$\begin{aligned} \hat{d}^2_{\mathrm{HE}}(P, Q | \alpha_x, \beta_y) &= \sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \alpha_x \alpha_{x'} \kappa(x, x') \zeta(x, x') + \sum_{y \in \mathcal{X}} \sum_{y' \in \mathcal{X}} \beta_y \beta_{y'} \kappa(y, y') \zeta(y, y') \\ &- 2 \sum_{x \in \mathcal{X}} \sum_{y' \in \mathcal{X}} \alpha_x \beta_y \kappa(x, y) \zeta(x, y). \end{aligned} \tag{8}$$

Then, we can write the equation (8) in matrix form as follows

$$\hat{d}^2_{\mathrm{HE}}(P_n, P_{n'}) = \boldsymbol{\alpha}_n^\top \boldsymbol{\kappa}^{n,n} \boldsymbol{\alpha}_n - 2 \boldsymbol{\alpha}_n^\top \boldsymbol{\kappa}^{n,n'} \boldsymbol{\alpha}_{n'} + \boldsymbol{\alpha}_{n'}^\top \boldsymbol{\kappa}^{n',n'} \boldsymbol{\alpha}_{n'}, \tag{9}$$

where $\boldsymbol{\alpha}_n$ is a vector of weights where each entry represents the weighting of each sample

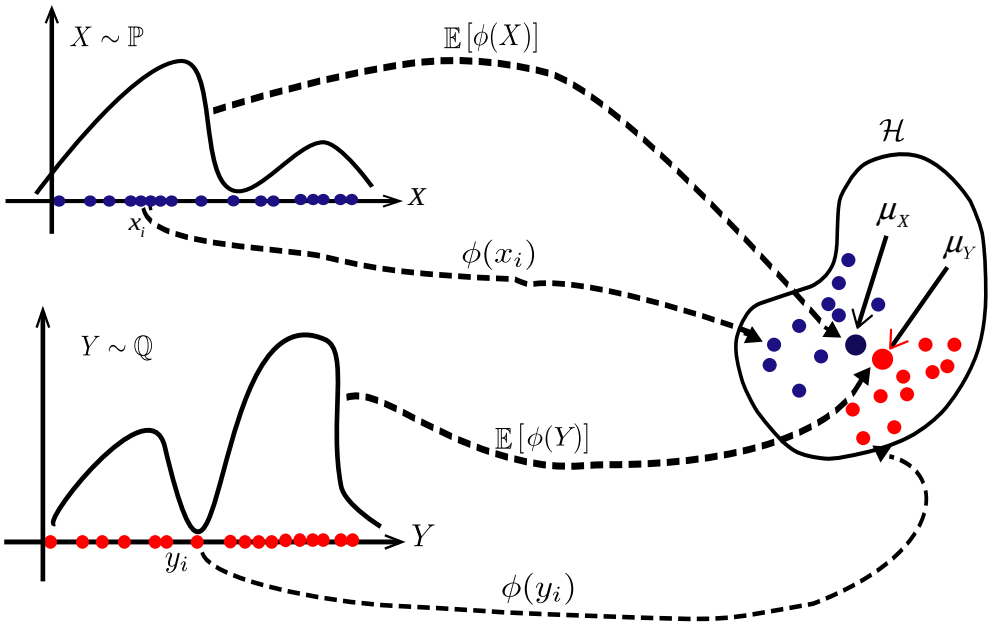


Figure 1: Mapping of probability distributions P and Q to an RKHS $\mathcal{H}$ from a fine set of samples

x' and $\boldsymbol{\kappa}^{n,n'}$ represent the kernel characteristic matrix between P_n and $P_{n'}$ distributions. To compute the expression (9), we must estimate the free hyperparameter of the characteristic kernel and the weights $\boldsymbol{\alpha}$. Weights can be estimated using different machines according to the application. We can use some of the algorithms most used in the state-of-the-art to model time series, such as HMMs, KAFs, SVMs, GPs, and others. In this work we show some aplications using particularly KAFs and HMMs machines.

7.2.1 Maximum Mean Discrepancy MMD

Maximum Mean Discrepancy (MMD) distance measure is the most simple RKHS-based metric and was introduced by Gretton *et al.* in [8]. In the MMD distance, the distributions P and Q have assumed empirics, it means

$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N}\delta(x - x'), \qquad \hat{q}(y) = \frac{1}{M}\sum_{m=1}^{M}\delta(y - y'),$$

where $\delta(\cdot)$ is the Delta distribution, N and M is the number of samples from distributions P and Q, respectively. Hence, according to (8), the RKHS-based MMD distance estimation is given by

$$\hat{d}^2_{\mathrm{HE}}(P,Q) = \frac{1}{N^2}\sum_{n,m=1}\zeta(x,x') + \frac{1}{M^2}\sum_{n,m=1}\zeta(y,y') - \frac{2}{NM}\sum_{n,m=1}\zeta(x,y),$$

where ζ is the kernel function defined in 7.2. Note that according to equation (8) $\alpha_x = 1/N$ and $\beta_y = 1/M$, it means that in the MMD distance it is assumed that all samples have the same relevance.

7.3 Hidden Markov Models (HMMs)

Formally, an HMM models a sequence of observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ by assuming that the observation at index i (i.e $\mathbf{x}_i$) was produces by an emission process associated to the k-valued discrete hidden state h_i and that sequences of hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$ was produces by a first-order Markov process. Therefore, the complete-data likelihood for a sequence of length n can be written as

$$p(\mathbf{X}, \mathbf{h}|\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(h_1|\boldsymbol{\pi})p(\mathbf{x}_1|h_1, \boldsymbol{\theta})\prod_{i=2}^{n}p(h_i|h_{i-1}, \mathbf{A})p(\mathbf{x}_i|h_i, \boldsymbol{\theta})$$

where $\mathbf{A} = \{a_{j,j'}\}$ denotes the hidden state transition matrix, $\boldsymbol{\pi} = \{\pi_j\}$ is the initial hidden state probability mass function and $\boldsymbol{\theta}$ represents the set of emission parameters for each hidden state. The problem of how estimate the HMM parameters $\boldsymbol{\phi} = \{\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\theta}\}$ is well-known and solutions for particular choices of emision processes have been proposed [5] [2].

7.4 Quantized Kernel Least Mean Square(QKLMS)

Kernel adaptive filters (KAFs) implement a nonlinear transfer function using kernel methods [27]. In these methods, the signal is mapped to a high-dimensional linear feature space and a

nonlinear function is approximated as a sum over kernels, whose domain is the feature space. If this is done in a reproducing kernel Hilbert space, a kernel method can be a universal approximator for a nonlinear function.

The quantized kernel least mean square (QKLMS) is a particular KAFs that take account a quantization method which compress the input space $\mathbf{u} \in \mathcal{U} \subseteq \mathcal{R}^m$. The QKLMS algorithm can be obtained by just quatizing the feature vector $\boldsymbol{\varphi}(i)$ in the weight-update equation $\boldsymbol{\Omega}(i) = \boldsymbol{\Omega}(i-1) + \eta e(i)\boldsymbol{\varphi}(i)$, which can be expressed as

$$\begin{cases} \boldsymbol{\Omega}(0) = \mathbf{0} \\ e(i) = d(i) - \boldsymbol{\Omega}(i-1)^\top \boldsymbol{\varphi}(i) \\ \boldsymbol{\Omega}(i) = \boldsymbol{\Omega}(i-1) + \eta e(i) \mathcal{Q}[\boldsymbol{\varphi}(i)] \end{cases} \tag{10}$$

where $\mathcal{Q}[\cdot]$ denotes a quantization operator in an inner product space $\mathcal{F}$ and $d = f(\mathbf{u})$ is the output signal. Since the dimensionality of the feature space is very high, the quantization is usually performed in the original input space $\mathcal{U}$. In this situation, the learning rule for QKLMS is

$$\begin{cases} f_0 = 0 \\ e(i) = d(i) - f_{i-1}(\mathbf{u}(i)) \\ f_i = f_{i-1} + \eta e(i) \zeta(Q[\boldsymbol{u}(i)], \cdot) \end{cases} \tag{11}$$

where $Q[\cdot]$ is a quantization in $\mathcal{U}$ and ζ is a Gaussian kernel with hyperparameter σ_x . More information about QKLMS filters can be found in [28].

7.5 The HSD and KL measures

The measures between HMMs proposed in this work are compared with the HMMs stationary distance (HSD) and Kullback-Leibler divergence (KL) . We follow the definition from [29]: if ρ_1 and ρ_2 are two HMMs and $\mathbf{X} = (X_1, X_2, \cdots, X_T)$ is a sequence of length T generated by ρ_1, then

$$D(\rho_1, \rho_2) = \frac{1}{T}\left(\sum_{i=1}^{T} \log p(x_i) - \log q(x_i)\right),$$

is the KL measure where p and q are the probability functions of ρ_1 and ρ_2, respectively, and they are estimated as in Equation (14). To achieve a more reasonable KL measure in

the experiments, KL is computed with the symmetric version as follows,

$$KL\left(\rho_1, \rho_2\right) = \frac{1}{2}\left(D\left(\rho_1, \rho_2\right) + D\left(\rho_2, \rho_1\right)\right)$$

The HSD measure is defined as in [29]: if ρ_1 and ρ_2 are two HMMs, then

$$HSD\left(\rho_1, \rho_2\right) = \int_{\mathcal{X}} \left|F_1(x) - F_2(x)\right| dx,$$

where $F_1(x) = \int_{\mathcal{X}} p(x)dx$ and $F_2(x) = \int_{\mathcal{X}} q(x)dx$ with p and q are stationary distributions of ρ_1 and ρ_2 (respectively), and they are estimated as in Equation (14). The HSD metric is calculated numerically.

8 Materials and Methods

8.1 RKHS-based distance using QKLMS

In this case, we propose to learn the time series in (6) towards a kernel adaptive filtering (KAF) technique. Namely, the quantized kernel least mean square (QKLMS) algorithm is selected as a straightforward solution [28]. So, the densities are computed from an input-output pair of samples $\{x_{n+1}, x_n\}_{n=1}^{N-1}$ and QKLMS predictions: $\hat{x}_{n+1} = \sum_{x' \in \Omega_n} \beta_{x'} \zeta(x_{n+1}, x')$, being Ω_n the filter codebook at the n-th iteration and $\beta_{x'} \in \mathbb{R}$. Therefore, QKLMS estimates the time serie into a RKHS following a Markovian constraint to preserve the temporal dependencie. Besides, the QKLMS includes a novelty criterion (NC) based on the euclidean distance $d_E(x_n, \Omega_{n-1})$, with $i \in \{1, 2, \ldots, N\}$. Later, given two time series from X and Y, that is, $\{x_n\}_{n=1}^{N} \sim P$ and $\{y_m\}_{m=1}^{M} \sim Q$, the densities are computed as:

$$\hat{p}(x) = \sum_{x' \in \Omega_{N-1}^{x}} \hat{\alpha}_{x'} \zeta(x, x') \qquad \hat{q}(y) = \sum_{y' \in \Omega_{N-1}^{y}} \hat{\beta}_{y'} \zeta(y, y'). \tag{12}$$

Furthermore, to preserve the metric properties in (9), the weights are normalized as: $\alpha_{x'} = |\beta_{x'}| / \sum_{\beta_{x'} \in \Omega_{N-1}^{x}} |\beta_{x'}|$ and $\beta_{y'} = |\beta_{y'}| / \sum_{\beta_{y'} \in \Omega_{M-1}^{y}} |\beta_{y'}|$. Finally, the equation (8) can be written as

$$\begin{aligned} \hat{d}_{HE}^{2}(\mathbb{P}, \mathbb{Q} | \alpha_x, \beta_y) &= \sum_{x, x' \in \Omega_{N-1}^{x}} \alpha_x \alpha_{x'} \kappa_G(x, x' | \sigma + 2\sigma_x) \\ &+ \sum_{y, y' \in \Omega_{M-1}^{y}} \beta_y \beta_{y'} \kappa_G(y, y' | \sigma + 2\sigma_y) - 2 \sum_{x \in \Omega_{N-1}^{x}, y \in \Omega_{M-1}^{y}} \alpha_x \beta_y \kappa_G(x, y | \sigma + \sigma_x + \sigma_y), \end{aligned} \tag{13}$$

where $\sigma, \sigma_x, \sigma_y \in \mathbb{R}^+$ are the bandwidth values for the characteristic and the QKLMS kernels, respectively. We have named (13) as HE-QKLMS.

8.2 RKHS-based distance using HMMs

In this work, we aim is classify time series from their representation in distributions from stationaries HMMs. According to equation (7) , the correspondings estimators of HMM stationary distributions p and q are given as

$$\hat{p}(x) = \sum_{i=1}^{N} \pi_{s,i}^{P} \widehat{b_i^P}(x), \quad \text{and} \quad \hat{q}(y) = \sum_{i=1}^{M} \pi_{s,i}^{Q} \widehat{b_i^Q}(y), \tag{14}$$

where $\pi_{s,i}^P$ and $\pi_{s,j}^Q$ are the stationary probabilities of the probability distributions P and Q. Since the emission probabilities are given by GMMs, we obtain

$$\widehat{b_i^P}(x) = \sum_{j=1}^{H_P} \delta_{i,j}\mathcal{N}(x|\mu_{i,j}, \Sigma_{i,j}), \quad \text{and} \quad \widehat{b_i^Q}(y) = \sum_{j=1}^{H_Q} \gamma_{i,j}\mathcal{N}(y|\nu_{i,j}.\Lambda_{i,j}),$$

Here, $\beta_{i,j}$ is the prior probability, $\nu_{i,j}$ is the mean parameter, and $\Lambda_{i,j}$ is the variance parameter for the component j of state i. Then, the mean parameter for the component j of state i and $\Lambda_{i,j}$ is the variance parameter for the component j of state i. Then, the mean maps for the distributions are given by

$$\widehat{\mu}_X(\mathbb{P}) = \int_{\mathcal{X}} k(\cdot, x)\widehat{p}(x)dx, \quad \text{and} \quad \widehat{\mu}_Y(\mathbb{Q}) = \int_{\mathcal{X}} k(\cdot, y)\widehat{p}(y)dy. \tag{15}$$

Now, replacing the expressions from Equation (15) in (6), and assuming a characteristic kernel $k(x, y; \ell)$, where ℓ is known as the bandwidth, then we obtain the RKHS-based distance between the distributions $\mathbb{P}$ and $\mathbb{Q}$ given by

$$\begin{aligned}\hat{d}_{HE}^2(P, Q) = &\sum_{i,j=1}^{N_P}\sum_{k,l=1}^{M_P} \pi_{s,i}^P\pi_{s,j}^P\alpha_{i,k}\alpha_{j,l}\widehat{\kappa}\left(\mu_{i,k}, \mu_{j,l}; \Sigma_{i,k}, \Sigma_{j,l}, \ell\right) \\ &+ \sum_{i,j=1}^{N_Q}\sum_{k,l=1}^{M_Q} \pi_{s,i}^Q\pi_{s,j}^Q\beta_{i,k}\beta_{j,l}\widehat{\kappa}\left(\nu_{i,k}, \nu_{j,l}; \Lambda_{i,k}, \Lambda_{j,l}, \ell\right) \\ &- 2\sum_{i,j=1}^{N_P,N_Q}\sum_{k,l=1}^{M_P,M_Q} \pi_{s,i}^P\pi_{s,j}^Q\alpha_{i,k}\beta_{j,l}\widehat{\kappa}\left(\mu_{i,k}, \nu_{j,l}; \Sigma_{i,k}, \Lambda_{j,l}, \ell\right).\end{aligned} \tag{16}$$

If the kernel $\kappa(x, y|\sigma) = \exp\left(-\sigma\|x - y\|_2^2\right)$ is Gaussian, then

$$\widehat{\kappa}\left(x, y; \Sigma, \Lambda, \sigma\right) = \frac{\sqrt{\sigma}}{\sqrt{\Sigma + \Lambda + \sigma}} \exp\left(-\frac{(x-y)^2}{2(\Sigma + \Lambda + \sigma)}\right), \tag{17}$$

where Σ, Λ, σ must be previously estimated. We have named (16) as HE-HMM.

8.3 Parameters estimation

This section describes the methods and strategies used to tune the parameters associated with the RKHS-based metrics, specifically the parameters of the KAFs, the HMMs and the characteristic kernel hyperparameter σ.

8.3.1 QKLMS parameters estimation

With respect to the QKLMS parameters, a grid of 100 values is build, varying $\epsilon_{\mathbb{U}}$ from 1 to 10, while e_t is computed based on the input data standard deviation. The kernel bandwidth σ_x is fixed according to a Parzen-based density estimation of each curvature sequence. Further, the characteristic kernel bandwidth σ is tuned in terms of the classification accuracy, building a grid of 5 points from 0.003 to 0.3 in *99-Shape Database*, and from 0.001 to 0.03 in *MPEG-7_CE-Shape-1 Part B*. The QKLMS learning rate is set as 0.9 for both experiments [27].

8.3.2 HMMs parameter estimation

The problem of estimating the model parameters $\boldsymbol{\phi} = \{\mathbf{A}, \boldsymbol{\pi}, \boldsymbol{\theta}\}$ given a sequence of observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is addressed in this section. A commonly used approach for training an HMM consists in choosing the parameters $\boldsymbol{\phi}$ in such a way that $p(\mathbf{X}|\boldsymbol{\phi})$ is maximized. This is known as a maximum-likelihood estimate for $\boldsymbol{\phi}$.

It turns out that the form of $p(\mathbf{X}|\boldsymbol{\phi})$ can not be directly maximized in a closed-form way. We then rely on the Expectation-Maximization (EM) algorithm [30] to estimate $\boldsymbol{\phi}$ which is an iterative procedure used for parameter estimation of probabilistic models with latent variables.

8.3.3 Characteristic kernel hyperparameter estimation

According to equation (9), the RKHS-based metrics depend to characteristic kernel that in turn depends on the σ hyperparameter. In general, this parameter is a matrix whose size depends on the dimensionality of the data. In the case of univariate time series this matrix is given as σI, where I is the identity matrix. However hyperparameter which must be accurately tuned for estimating an RKHS. Otherwise, a wrong bandwidth value leads to distinct not fulfilling the learning task. In most applications, the characteristic kernel hyperparameter is adjusted based on cross-validation heuristic techniques [20] [22] [21]. These approaches require the construction of a grid of possible values in order to evaluate the performance obtained for each one, which can lead to inaccurate values because the optimal value may not be contained in the grid. Also, the cross-validation is a procedure with high computational cost [31].

In this work, we propose two automatic selections of the characteristic kernel hyperparameter of the RKHS-based metrics [8]. The main goal is to construct a suitable RKHS for time series classification. Inspired by the approach [32], we use the potential information variability

(IPV) from a Parzen-based probability density estimator as an unsupervised estimation method and acording [33] we use a metric learnig supervised method for the hyperparameter estimation.

8.3.4 Kernel Function Estimation from Information Potential Variability

In order to estimate the characteristic kernel σ hyperparameter of (9), we used an automatic selection strategy based on IPV developed in [32]. The method is described as follows: let $\mathbf{\Lambda}$ a set of observable data, then a density function $p(\boldsymbol{\lambda}, \sigma)$ is estimated using a Gaussian Parzen-window density estimator, that is:

$$\hat{p}(\boldsymbol{\lambda}, \sigma) = \frac{1}{N_T} \sum_{i,j=1}^{N_T} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(\frac{\|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\|^2}{2\sigma^2}\right), \tag{18}$$

where $\boldsymbol{\lambda} \subset \mathbf{\Lambda}$, D is the dimensionality of the input space and N_T is the total number of samples. In this sense, $\hat{p}(\boldsymbol{\lambda}, \sigma)$ depends on a Gaussian kernel and can be written concerning the Euclidean distance as seen in equation (18). Then, we seek an RKHS maximizing the overall (IPV) respect to σ. To this end, the variability of $\hat{p}(\boldsymbol{\lambda}, \sigma)$ is maximized in terms of kernel bandwidth parameter as:

$$\sigma^* = \arg\max_{\sigma} \operatorname{var}\{\hat{p}(\boldsymbol{\lambda}, \sigma)\}, \tag{19}$$

where $\operatorname{var}\{\hat{p}(\boldsymbol{\lambda}, \sigma)\} = \mathbf{E}\{(\hat{p}(\boldsymbol{\lambda}, \sigma) - \mathbf{E}\{\hat{p}(\boldsymbol{\lambda}, \sigma)\})^2\}$. Deriving (19) with respect to σ, the optimal parameter value can be written in terms of information potential (IP) $V(\mathbf{\Lambda})$ and information force (IF) $F(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j})$ as

$$\frac{d}{d\sigma}\operatorname{var}\{\hat{p}(\boldsymbol{\lambda}, \sigma)\} = \frac{2(N_T^2 + N_T)}{\sigma} \left(\sigma^2 \sum_{i,j=1}^{N_T} F^2(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j}) - V(\mathbf{\Lambda}) \sum_{i,j=1}^{N_T} (F(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j}))^\top (\boldsymbol{\lambda_i} - \boldsymbol{\lambda_j}) \right).$$

Finally, equaling the above equation to zero, the fixed point update rule becomes:

$$\sigma_{k+1}^2 = \frac{V_k(\mathbf{\Lambda}) \mathbf{E}\{(F_k(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j}))^\top (\boldsymbol{\lambda_i} - \boldsymbol{\lambda_j}) : \forall i, j \in [1, N_T]\}}{\mathbf{E}\{F_k^2(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j}) : \forall i, j \in [1, N_T]\}}, \tag{20}$$

where $V_k(\mathbf{\Lambda})$ and $F_k(\boldsymbol{\lambda_i}|\boldsymbol{\lambda_j})$ are the IP and conditional IF obtained when $\sigma = \sigma_k$, respectively. In this way, we obtained a scale rule as a function of the IFs, which are induced by a kernel applied over a finite sample set. This approach is named for authors as Kernel Function Estimation from Information Potential Variability (KEIPV) [32]. Notably, the optimization problem described in the equation (19) is non-convex, that is, the σ value may converge to a local minimum. Therefore, the performance of the optimization process may be affected unless it is initialized suitability.

8.3.5 Centered Kernel Alingment

In this section we describe an alternative approache to tune the characteristic kernel hyperparameter of (9) based on supervised learning, especifically we use and metric learning algorithm knowledge as Centered Kernel Alingment (CKA) [33] [34]. The basic idea consist in to maximize a similarity measure between kernels or kernel matrices. The kernels we want to alingn are the centered version of the kernel matrix $\mathbf{l}$ computed from the labels, and the centered version of the characteristic kernel matrix $\boldsymbol{\kappa}$. The centered kernel matrices corresponding to $\mathbf{l}$, and $\boldsymbol{\kappa}$ are given by $\tilde{\mathbf{l}} = \mathbf{HlH}$, and $\tilde{\boldsymbol{\kappa}} = \mathbf{H}\boldsymbol{\kappa}\mathbf{H}$, respectively, with $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. Hyperparameter σ, or in general Σ matrix, for multidimensional times series, is obtained solving

$$\Sigma^* = \arg\max_{\Sigma}[\log \rho(\mathbf{l}, \boldsymbol{\kappa})], \tag{21}$$

where

$$\rho(\mathbf{l}, \boldsymbol{\kappa}) = \frac{\langle \tilde{\mathbf{l}}, \tilde{\boldsymbol{\kappa}} \rangle_F}{||\tilde{\mathbf{l}}||_F ||\tilde{\boldsymbol{\kappa}}||_F}. \tag{22}$$

In the expression above $\langle \cdot, \cdot \rangle_F$ stands for the Frobenius product, and $||\cdot||_F$ for the Frobenius norm. The Frobenius product between matrices $\mathbf{A} \in \mathcal{R}^{N \times N}$, and $\mathbf{B} \in \mathcal{R}^{N \times N}$ is defined by

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \mathrm{tr}(\mathbf{A}^\top \mathbf{B}), \text{ and } ||\mathbf{A}||_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F}. \tag{23}$$

Expression for $\log \rho(\mathbf{l}, \boldsymbol{\kappa})$ can also written as

$$\log \rho(\mathbf{l}, \boldsymbol{\kappa}) = \log \mathrm{tr}(\mathbf{lH}\boldsymbol{\kappa}\mathbf{H}) - \log \sqrt{\mathrm{tr}(\boldsymbol{\kappa}\mathbf{H}\boldsymbol{\kappa}\mathbf{H})} - \log \sqrt{\mathrm{tr}(\mathbf{lHlH})}, \tag{24}$$

where we have used the property $\langle \tilde{\mathbf{l}}, \tilde{\boldsymbol{\kappa}} \rangle_F = \langle \mathbf{l}, \tilde{\boldsymbol{\kappa}} \rangle_F = \langle \tilde{\mathbf{l}}, \boldsymbol{\kappa} \rangle_F$. Expresion for Σ^* can then be written as

$$\Sigma^* = \arg\max_{\Sigma}[\log \mathrm{tr}(\mathbf{lH}\boldsymbol{\kappa}\mathbf{H}) - \log \sqrt{\mathrm{tr}(\boldsymbol{\kappa}\mathbf{H}\boldsymbol{\kappa}\mathbf{H})} - \log \sqrt{\mathrm{tr}(\mathbf{lHlH})}]. \tag{25}$$

Finally, to obtain Σ^*, we use a gradient-descend algorithm.

9 Experiments

In this section the capability of the proposed methodology for learning time series and classifying them is evaluated. In the first part this validation is done using synthetic and real datasets.

9.1 Shape classification

Here, we introduce a shape classification approach based on a curvature-based representation and an RKSH-based metric HE-QKLMS described in (13). Namely, a kernel adapting filtering (KAF) strategy is carried out to learn salient patterns from curvature-based features [27]. In turn, such patterns are used to compute the probability density of each object shape from a RKHS-based perspective. Lastly, a k-nearest neighbors classifier is trained using the HE-QKLMS metric to discriminate shapes from binary images. As a benchmark, we employ the stochastic model hinge on likelihood maximization presented in [35], and a straightforward MMD algorithm [36]

The binary images preprocessing is described as follows: given a binary image $M \in \mathbb{R}^{W \times H}$ holding $W \times H$ pixels, a Canny filter is applied as edge detector to reveal the object shape. Then, some morphological operations are carried out to prevent non-closed trajectories and a set of curvature coefficients $\{x_n \in \mathbb{R}\}_{n=1}^{N}$ is extracted based on first order changes. In fact, the trajectories are extracted from farthest horizontal point to the right, getting each curvature point from this first coordinate in a counter-clockwise manner.

To assess the HE-QKLMS metric we used two databases (DB): the *99-Shape Database* [1], and the *MPEG-7_CE-Shape-1 Part B* [2], holding 9 and 13 classes, respectively (see figure 2). Both datasets comprise binary images with different resolutions and perturbations, e.g., occlusion and rotation. To get the shape curvatures we replicate the feature estimation methodology presented by authors in [2], tuning experimentally the length parameter to compute the curvature at each spatial point. We implement a 1-NN classifier from the RKHS-based distances among shapes. For concrete testing, a training-testing validation is carried out fixing the training set as the 80% and 50% of the input images for *99-Shape Database MPEG-7_CE-Shape-1 Part B*, respectively. QKLMS parameter were estimated according to secction 8.3.1. As baseline, we test two state-of-the art techniques: a HMM classifier based on highest likelihood as discussed in [35], and a 1-NN classifier from a MMD representation [36] using equation (10). Figure 3 left, shows a data sample contour plot from *99-Shapes* DB. As seen, points holding high curvature coefficient values (see color intensities)

[1] http://vision.lems.brown.edu/content/

[2] http://www.dabi.temple.edu/~shape/MPEG7/dataset.html

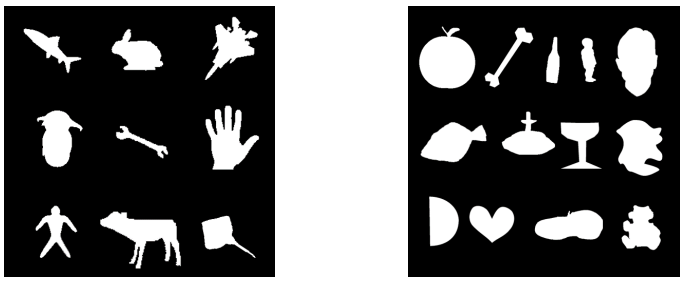


Figure 2: Left: 99-Shapes DB which has 11 samples per class. Right: MPEG-7 DB which has 20 samples per class.

are related to extremities, whose has greater morphological changes concerning the rest of the shape. Besides, from marker sizes, the QKLMS-based relevance analysis is able to code the main shape variations. Now, Figure 3 right shows the filter prediction against the target curvature sequence. Curvature beginning is labeled by x_0, and as was exposed previously, following counter-clockwise it is found the midpoint of the sequence, labeled as x_{40}. In general, it is clear that the KAF-based adjustment is low concerning the curvature predictions. Since our goal is to classify shapes towards a relevant curvature representation, the codebook and the filter weights are more appropriate for further discrimination task, e.g., the 1-NN-based classification, than for the curvature value prediction. So, the training stage requires a trade-off between classification and filtering fitting. In addition, Table 1 shows the results

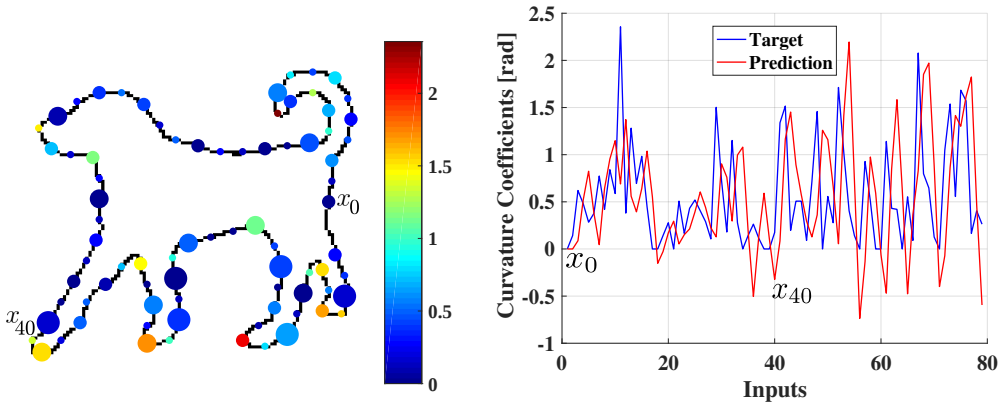


Figure 3: Left: $\{x_n\}_{n=1}^{N}$ for image 63 in the 99-Shape, where color indicates a x_n value and marker size code the QKLMS-based relevance. Right: QKLMS curvature prediction.

of classification for the *99-Shapes* DB. Overall, the accuracy obtained in this experiment was 77.8%, which is lower than the HMM and MMD approaches. However, in Quadrupeds, Humans, Hands, Rays, Rabbits, and Wrenches classes, our proposal achieves the highest performances. Moreover, for the Airplanes class, our methodology fails; this is ought to sharp

changes along its contour and low curvature sequences in comparison to the other classes, which induces biased prediction results in the QKLMS filter. Further, Table 2 presents the classification results for the *MPEG-7_CE-Shape-1* DB. This experiment was successful since our methodology overcomes MMD and HMM schemes. However, for Shoe and Bone classes, their outcomes could be due to their class heterogeneity. For Bottle, Children, Flatfish, and Fountain classes, our proposal allows discriminating each of them properly.

In general, our HE-QKLMS distance is competitive with other state-of-the-art methodologies. Previous results allow us to say that our methodology performs better than HMM models proposed by [35]. This outcome could be possible ought to our method introduces the NC (see Section 7.4) [28], which avoids using redundant curvature sequences. Moreover, these input filters are embedded into an RKHS, which allows to maps nonlinear shape structures.

Table 1: *99-Shapes* classification results.

Class	HE-QKLMS TP(%)	HMM TP(%)	MMD TP(%)
Quadrupeds	100.0	100.0	50.0
Humans	100.0	100.0	100.0
Airplanes	0.0	81.8	50.0
Grebes	50.0	100.0	50.0
Fish	50.0	72.7	100.0
Hands	100.0	90.9	100.0
Rays	100.0	90.9	100.0
Rabbits	100.0	81.8	100.0
Wrenches	100.0	72.7	100.0
Accuracy(%)	**77.8**	87.9	83.3

Table 2: *MPEG-7* classification results.

Class	HE-QKLMS TP(%)	HMM TP(%)	MMD TP(%)
Bone	60.0	100.0	60.0
Glas	100.0	100.0	100.0
HCircle	90.0	80.0	80.0
Heart	90.0	90.0	70.0
Misk	100.0	80.0	100.0
Apple	70.0	80.0	60.0
Bottle	100.0	70.0	90.0
Children	100.0	90.0	90.0
Face	100.0	90.0	90.0
Flatfish	100.0	80.0	90.0
Fountain	100.0	80.0	90.0
Shoe	50.0	60.0	90.0
Teddy	80.0	90.0	100.0
Accuracy(%)	**87.7**	83.8	85.4

9.2 Automatic Assessment of Voice Quality

A technique that has been developed to detect pathologies associated with the voice quality is the acoustic analysis. This technique is non-invasive and is based on the digital processing of the speech signal. Through this processing, a time series or spectral features can be extracted from the voice signal, which is supposed to be related to its quality [**?**, **?**]. In this

section, we evaluate the performance of the HE-HMM metric described in (16) concerning the Kulback-Leibler (KL) divergence and the Dynamical Time Warping (DTW) algorithm in a voice database. The voice database used in this article was the Massachusetts Eye and Ear Infirmary Disordered Voice Database from the Kay Elemetrics company. Specifically, the subset of 226 voice records described by [**?**].

Description of the voice database: the dataset contains the characterization of 226 voice samples from the cepstral coefficients in the Mel frequency scale. Each voice sample is divided into windows with a duration of ten milliseconds. Finally, each window is characterized using: twelve cepstral coefficients, an energy term, the first derivative of the coefficients and the second derivative of the coefficients. In this sense, each voice window will be parameterized by a total of thirty-nine characteristics, and each signal belongs to one of the two classes, pathological voice or a healthy voice. Finally, in this experiment, we use a 1-NN classifier again with HE-HMM, KL, and DTW distances. We configured the HMMs as follows section 8.3.2. For DTW-based classifier, we used the originals signals speech.

To test the statistical significance, we realized ten repetitions of each one of the classifiers using Hold-out partitions. According to Table 3 it is observed that the HE-HMM metric has better accuracy in classification than DTW measure for the voice database. In another hand, HE-HMM obtained similar performance regarding KL. To asses the statistical signicance between the different measures, we applied a Lilliefors test for normality over the ten repetitions of each classifier. If the null hypothesis for normality is rejected, we perform a Kruskal-Walls test to compare average performances among the classifiers. If the null hypothesis for equal medians is rejected, we perform multiple comparison tests using Tukey-Kramer to study further wich classifiers are different. All the significance levels are measured at 5% [37]. We found that the accuracy results of the HE-HMM and DTW metrics are statistically different, and results for the HE-HMM and KL metrics are statistically equal. Probably a reason for the better performance of the HE-HMM metric with respect to the DTW measure is that this last measure is not suited to multichannel signal speech recognition [**?**].

Table 3: *Accuracy results using the HE-HMM, KL and DTW metrics for* $K = 1$. *The mean* μ *and the standard deviation* σ *are shown for ten reps of each experiment* ($\mu \pm \sigma$).

$HE-HMM$	KL	DTW
$\mathbf{0.8745 \pm 0.0100}$	0.8679 ± 0.0267	0.8481 ± 0.0196

9.3 UCR data repository

In this section we show the results of testing our proposed methodology on public datasets from to *University of California in Riverside (UCR) time series classification repository.*

Then, we used thirty-one binary datasets corresponding to synthetic and real-world problems. Each one of datasets come previously partitioned in train and test datasets. Table 4 shows the sizes of the training and testing sets and the length of the time series. We test our proposals selection methods for characteristic kernel hyperparameters IPV and CKA for HE-HMM and MMD RKHS-based metrics. First, to show the performance of IPV approach

Table 4: *The thirty-one binary databases with the size of training set and size of testing set, used in this paper to compare the performance of the distance measures proposed.*

Dataset	Train size		Test size		Length
DistalPhalanxOutlineCorrect	115	161	222	378	80
ShapeletSim	10	10	90	90	500
ToeSegmentation1	20	20	120	108	277
Computers	125	125	125	125	720
Herring	39	25	38	26	512
Ham	52	57	51	54	431
Wine	30	27	27	27	234
Wafer	97	903	665	5499	152
Coffee	14	14	15	13	286
Strawberry	132	238	219	394	235
ECGFiveDays	14	9	428	433	136
MoteStrain	10	10	675	577	84
ProximalPhalanxOutlineCorrect	194	406	92	199	80
BirdChicken	10	10	10	10	512
Earthquakes	104	35	264	58	512
SonyAIBORobotSurfaceII	11	16	365	588	953
ItalyPowerDemand	34	33	513	516	24
Lightning2	20	40	28	33	637
ECG200	31	69	36	64	96
GunPoint	24	26	76	74	150
BeetleFly	10	10	10	10	512
SonyAIBORobotSurface	6	14	343	258	601
PhalangesOutlinesCorrect	628	1172	332	526	80
TwoLeadECG	12	11	569	570	82
MiddlePhalanxOutlineCorrect	125	166	212	388	80
ToeSegmentation2	18	18	106	24	343
WormsTwoClass	33	44	76	105	900
HandOutlines	133	237	362	638	2709
Yoga	137	163	1393	1607	426
FordA	681	639	1846	1755	500
FordB	401	409	1860	1776	500

for automatic selection of characteristic kernel hyperparameter described in section 8.3.4 , we test this method using the Wafer dataset from UCR repository. We use the training set $\mathbf{\Lambda} \in \mathbb{R}^{1000\times 152}$ as input space, then we projected $\mathbf{\Lambda}$ to $\mathbf{Z} \in \mathbb{R}^{1000\times 2}$ using Principal Components Analysis (PCA) representation. Thus, we represent time series as points in the projected space. Figure 4(a) shows that for $\sigma = 1 \times 10^{-3}$ low similarities between pair-wise of time series (particles) and low magnitude IFs are computed due to Gaussian kernel κ reduces the scaling of the Euclidean distance between particles. For this reason, particles are forced to apart from each other. In another hand, in Figure 4(b), we can see how IFs magnitudes change regardless their directions, that is, close particles according to the Euclidean distance get high pairwise similarities while far ones have low similarities using the σ^* parameter obtained through IPV. Therefore, IPV finds an RKHS where time series share

widely spread IF magnitudes. Figure 4(c) shows the accuracy in terms of σ values chosen

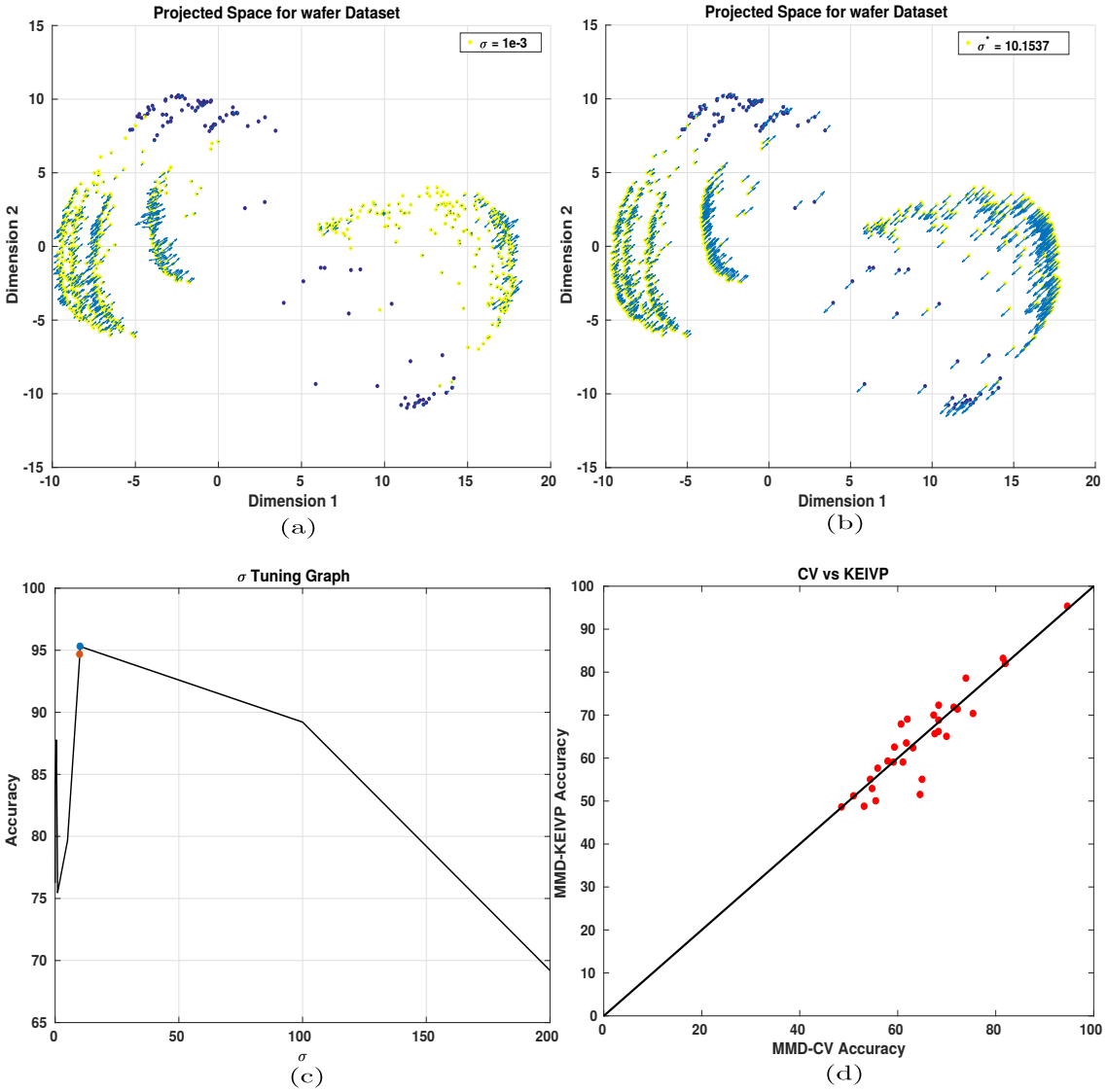
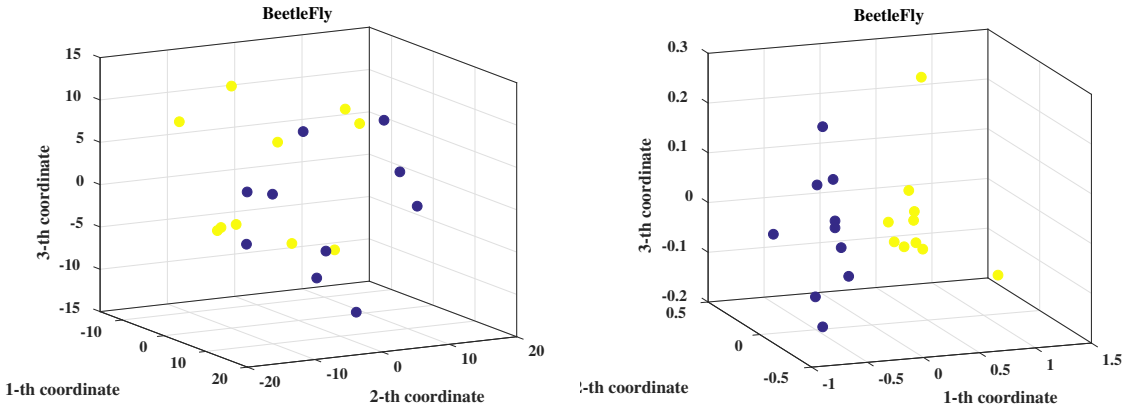


Figure 4: *Figure 4(a) illustrates the IFs when the $\sigma = 1 \times 10^{-3}$, in this case, particles tend to apart from each other and IPV is slow. Figure 4(b) shows that for the selected parameter using KEIVP, IFs magnitudes change regardless of their directions. Figure 4(c) shows the accuracy obtained using the CV approach, the blue point is the optimal using KEIVP methodology, and red point is the highest value obtained with CV. Figure 4(d) exhibits the comparison between our approach and CV, red points shows the accuracies values from columns 6 and 7 of Table 5. Points above black line represent the accuracies where our method is better than CV. Points below black line vice versa.*

as of cross-validation (CV) for Wafer dataset. The accuracy obtained using CV for Wafer

dataset was 94.66%, similar to accuracy obtained using IPV methodology (95.30%). We can see that the automatic selection of σ parameter using IPV achieves find an optimal value (blue point) close to the one that makes the best performance using CV (red mark). Besides, Table 5 shows the results of classification for thirty-one binary datasets from UCR repository. Overall, the mean of accuracies obtained using our methodology was $64,95\%$ (column 7), which is lower than the MMD-based 1-NN classifier using CV (column 6). However, our approach succeeded to win for five-ten datasets, while CV approache achieve to win over four-teen datasets. Figure 4(d) exhibits the comparison between our proposal and CV; red points show the accuracies values for the thirty-one binary datasets from UCR. Points above the black line represent the accuracies where our method is better than CV. Points below black line vice versa. For GunPoint and Ham datasets, both methodologies obtained equal performance (points on the black line). In addition, our methodology achieves improve the performance of CV when the proportion between the sizes of training and validation sets is similar (see Table 4). Otherwise, our approach achieves low performance when the training set is small with respect to the validation set because IPV takes advantage of the information provided by all training set $\mathbf{\Lambda}$ according to section 8.3.4. In general, MMD_IPV methodology is competitive with respect to MMD using CV. Previous results allow us to say that our approach achieves tune hyperparameter σ suitability, allows estimating RKHSs favoring data class separability in comparison with a heuristic way (cross-validation).

Now, we show the performance of automatic selection of characteristic kernel hyperparameter using an approache based-on CKA described in section 8.3.5. We test CKA method using the BeetleFly dataset from UCR repository. We use the training set $\mathbf{\Lambda} \in \mathbb{R}^{20\times 512}$ as input space, then we projected $\mathbf{\Lambda}$ to $\mathbf{Z} \in \mathbb{R}^{20\times 3}$ using a multi-scale representation based on Singular Value Decomposition (SVD). Thus, we represent time series as points in the projected



(a) *Multi-scale representation of original dissimilarity space.*

(b) *Multi-scale representation of rotated dissimilarity space using CKA.*

space. Figure 9.3(b) show that CKA achive rotate the original dissimilarity space (Figure

9.3(a)) getting a better representation in terms of separability of the samples and therefore facilitating the task of discrimination between these. The first five columns from Table 5 show the accuracy classification for the thirty-one binary data sets from UCR repository using 1-NN classifier based on defferents distance between HMMs. First and second column show results obtained using two baseline, KL and HSD. Third column show results using HE-HMM proposed metric tunning the characteristic kernel hyperparameter in a heuristic way (cross-val). Fourth and fifth column show HE-HMM using IPV and CKA methods of selection, respectively. We can see that the performance of our proposal is generally better than that of KL $(59, 84\%)$ and competitive with HSD $(68, 59\%)$. The last three columns show accuracy results using MMD distance with CV, IPV and CKA, respectively. As we mentioned earlier, IPV for kernel hyperparameter tuning, achive to find a suitable RKHS in classification terms. However, CKA-based strategy is better than IPV in terms of accuracy. That is, by rotating the representation space it is possible to guarantee better separability in the samples, which leads to better classification results (last column). In general MMD with CKA achive win over twenty of thirty-one datasets.

Table 5: *The thirty-one binary databases used to compare the performance of the RKHS-based metrics HE-HMM and MMD with respect to the KL and HSD measures using the 1-NN algorithm for thirty-one binary datasets from UCR repository. We test out methodology using IPV and CKA methods of selection for characteristic kernel hyperparameter.*

Dataset	KL	HSD	HE-HMM	HE-HMM_IPV	HE-HMM_CKA	MMD	MMD_IPV	MMD_CKA
BeetleFly	**85,00**	65,00	70,00	72,30	65,00	65,00	55,00	75,00
BirdChicken	75,00	**85,00**	**85,00**	85,53	80,00	70,00	65,00	80,00
Coffee	64,29	75,00	57,14	59,00	89,29	60,71	67,86	**92,86**
Computers	52,40	66,40	62,80	63,63	62,00	68,40	**72,25**	69,20
DistalPhalanxOutlineCorrect	59,67	66,50	62,67	61,45	74,50	59,17	59,00	**80,83**
ECG200	49,00	68,00	55,00	72,00	70,00	62,00	69,00	**76,00**
ECGFiveDays	60,16	79,44	75,84	74,42	77,82	72,24	71,34	**87,34**
Earthquakes	69,57	69,57	70,19	81,98	**81,99**	**81,99**	81,98	79,81
FordA	51,07	55,07	50,82	51,06	54,18	53,18	48,73	**55,57**
FordB	48,16	**54,98**	52,26	52,34	53,42	51,02	51,15	53,44
GunPoint	**91,33**	83,33	74,67	85,78	68,67	82,00	82,00	75,33
Ham	48,57	51,43	48,57	48,57	53,33	48,57	48,57	**68,57**
HandOutlines	61,10	63,10	59,10	62,30	60,40	**67,60**	65,60	65,62
Herring	51,56	45,31	40,62	55,98	57,81	59,38	62,50	**68,75**
ItalyPowerDemand	50,63	75,12	71,23	71,23	**88,82**	54,81	52,85	78,91
Lightning2	73,77	73,77	70,49	65,64	60,66	75,41	70,32	**78,69**
MiddlePhalanxOutlineCorrect	37,50	63,33	57,83	58,00	60,83	63,17	62,33	**73,67**
MoteStrain	75,24	74,92	72,76	70,43	71,96	73,96	77,15	**80,91**
PhalangesOutlinesCorrect	40,09	64,69	58,74	57,63	70,63	55,94	57,58	**73,89**
ProximalPhalanxOutlineCorrect	30,58	72,51	70,79	71,24	72,51	68,38	68,76	**80,07**
ShapeletSim	51,11	51,67	45,00	50,00	58,89	55,56	50,00	**79,44**
SonyAIBORobotSurface	**81,36**	74,88	66,39	61,37	59,57	67,39	69,95	74,71
SonyAIBORobotSurfaceII	72,30	64,64	56,03	62,46	75,97	61,80	63,45	**79,75**
Strawberry	48,45	76,84	69,33	69,33	82,22	68,35	66,13	**90,70**
ToeSegmentation1	70,61	72,81	69,74	70,47	67,11	71,49	71,78	**74,12**
ToeSegmentation2	50,77	68,46	60,77	61,32	63,85	81,54	**83,15**	69,23
TwoLeadECG	59,09	69,27	70,32	72,38	67,25	54,43	55,00	**73,84**
Wine	51,85	61,11	61,11	62,92	62,96	61,11	59,00	**68,52**
WormsTwoClass	51,38	62,43	58,01	59,00	60,22	58,01	59,24	**65,75**
Wafer	75,83	96,45	92,83	95,42	95,42	94,66	95,30	**97,16**
Yoga	**67,80**	75,23	67,70	58,97	59,13	64,60	51,46	56,93
Average	59,84	68,59	63,99	65,94	68,59	65,54	64,95	**74,99**

10 Conclusions and Future works

10.1 Conclusions

In this work a general distance measure between times series using RKHS embedding is proposed. We build, as special cases two metrics based on HMMs and QKLMS filter in order to code temporal dependencies in the data. Also, we propose two strategies of selection (IPV and CKA) of the most suitable RKHS in order to guarantee separability of the samples and thus obtain better classification results. We test our methodologies in different applications using synthetic and real data obtaining in most cases competitive results with respect to the other methods of the state of the art.

10.2 Research Outcomes

In terms of scientific production we achived:

- Blandon, J. S., Valencia, C. K., Alvarez, A., Echeverry, J., Alvarez, M. A., & Orozco, A. (2018, June). *Shape classification using Hilbert space embeddings and kernel adaptive filtering*. In International Conference Image Analysis and Recognition (pp. 245-251). Springer, Cham.
- Valencia, C. K., Álvarez, A., Valencia, E. A., Álvarez, M. A., & Orozco, Á. (2018, November). *Information Potential Variability for Hyperparameter Selection in the MMD Distance*. In Iberoamerican Congress on Pattern Recognition (pp. 279-286). Springer, Cham.
- Valencia, E. A., Valencia, C. K., Lopez-Lopera, A.F,& Álvarez, M. A. (2019) *Distance measures for hidden Markov models based on Hilbert space embeddings for time series classification*. Advances in Data Analysis and Classification. Springer (**submitted**)

10.3 Future Work

We consider the following possible ways of extending this work:

- To build an extension for RKHS-based metric using not stationary HMMs and GPs.
- In order to codify space-time dependencies, extend the proposed methodology for graphic models such as Random Marcov Fields.

- An alternative estimation for characteristic kernel hyperparameter based on Bayesian Optimization.

Bibliografía

[1] A. Kotsifakos, "Case study: Model-based vs. distance-based search in time series databases," in *Exploratory Data Analysis (EDA) Workshop in SIAM International Conference on Data Mining (SDM)*, 2014.

[2] M. Bicego and V. Murino, "Investigating hidden markov models' capabilities in 2d shape classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 281–286, 2004.

[3] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[4] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[5] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[6] J. C. Principe, *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[7] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[8] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A hilbert space embedding for distributions," in *ICALT*. Springer, 2007, pp. 13–31.

[9] P. Anantasech and C. A. Ratanamahatana, "Enhanced weighted dynamic time warping for time series classification," in *Third International Congress on Information and Communication Technology*. Springer, 2019, pp. 655–664.

[10] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, pp. 1–47, 2019.

[11] M. Bicego, V. Murino, and M. A. Figueiredo, "Similarity-based classification of sequences using hidden markov models," *Pattern Recognition*, vol. 37, no. 12, pp. 2281–2291, 2004.

[12] P. Tanisaro and G. Heidemann, "Time series classification using time warping invariant echo state networks," in *ICMLA, 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 831–836.

[13] M. Bicego, V. Murino, and M. A. Figueiredo, "A sequential pruning strategy for the selection of the number of states in hidden markov models," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1395–1407, 2003.

[14] A. Singh and J. C. Príncipe, "Information theoretic learning with adaptive kernels," *Signal Processing*, vol. 91, no. 2, pp. 203–213, 2011.

[15] D. R. PW and P. Elzbieta, *Dissimilarity Representation For Pattern Recognition, The: Foundations And Applications*. World scientific, 2005, vol. 64.

[16] V. R. Marco, D. M. Young, and D. W. Turner, "The euclidean distance classifier: an alternative to the linear discriminant function," *Communications in Statistics-Simulation and Computation*, vol. 16, no. 2, pp. 485–505, 1987.

[17] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[18] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[19] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 961–968.

[20] C. D. Zuluaga, E. A. Valencia, M. A. Álvarez, and Á. A. Orozco, "A parzen-based distance between probability measures as an alternative of summary statistics in approximate bayesian computation," in *International Conference on Image Analysis and Processing*. Springer, 2015, pp. 50–61.

[21] J. Blandon, C. Valencia, A. Alvarez, J. Echeverry, M. Alvarez, and A. Orozco, "Shape classification using hilbert space embeddings and kernel adaptive filtering," in *International Conference Image Analysis and Recognition*. Springer, 2018, pp. 245–251.

[22] W. González-Vanegas, A. Alvarez-Meza, and Á. Orozco-Gutierrez, "Sparse hilbert embedding-based statistical inference of stochastic ecological systems," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2017, pp. 255–262.

[23] M. M. Ramón, "Introducción a los métodos kernel," *Universidad Autónoma de Madrid*, vol. 29, 2008.

[24] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.

[25] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1517–1561, 2010.

[26] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[27] W. Liu, J. C. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*. John Wiley & Sons, 2011, vol. 57.

[28] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 22–32, 2012.

[29] J. Zeng, J. Duan, and C. Wu, "A new distance measure for hidden markov models," *Expert systems with applications*, vol. 37, no. 2, pp. 1550–1555, 2010.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[31] A. W. Moore and M. S. Lee, "Efficient algorithms for minimizing cross validation error," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 190–198.

[32] A. M. Álvarez-Meza, D. Cárdenas-Peña, and G. Castellanos-Dominguez, "Unsupervised kernel function building using maximization of information potential variability," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2014, pp. 335–342.

[33] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 795–828, 2012.

[34] A. J. Brockmeier, J. S. Choi, E. G. Kriminger, J. T. Francis, and J. C. Principe, "Neural decoding with kernel-based metric learning," *Neural computation*, vol. 26, no. 6, pp. 1080–1107, 2014.

[35] C.-M. Pun and C. Lin, "Geometric invariant shape classification using hidden markov model," in *DICTA, 2010 International Conference on*. IEEE, 2010, pp. 406–410.

[36] C. Luo and L. Ma, "Manifold regularized distribution adaptation for classification of remote sensing images," *IEEE Access*, 2018.

[37] H. D. V. Cardona, A. A. Orozco, and M. A. Alvarez, "Multi-patient learning increases accuracy for subthalamic nucleus identification in deep brain stimulation," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 4341–4344.