Pattern Recognition 84 (2018) 211–225

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A novel formulation of orthogonal polynomial kernel functions for SVM classifiers: The Gegenbauer family

Luis Carlos Padierna[a], Martín Carpio[a], Alfonso Rojas-Domínguez[a,1,*], Héctor Puga[a], Héctor Fraire[b]

[a] Tecnológico Nacional de México, Instituto Tecnológico de León, León 37290, México
[b] Tecnológico Nacional de México, Instituto Tecnológico de Cd. Madero, Cd. Madero 89460, México

ARTICLE INFO

Article history:
Available online 12 July 2018

Keywords:
SVM classifier
Orthogonal polynomials
Gegenbauer kernel
Binary classification

ABSTRACT

Orthogonal polynomial kernels have been recently introduced to enhance support vector machine classifiers by reducing their number of support vectors. Previous works have studied these kernels as isolated cases and discussed only particular aspects. In this paper, a novel formulation of orthogonal polynomial kernels that includes and improves previous proposals (Legendre, Chebyshev and Hermite) is presented. Two undesired effects that must be avoided in order to use orthogonal polynomial kernels are identified and resolved: the Annihilation and the Explosion effects. The proposed formulation is studied by means of introducing a new family of orthogonal polynomial kernels based on Gegenbauer polynomials and comparing it against other kernels. Experimental results reveal that the Gegenbauer family competes with the RBF kernel in accuracy while requiring fewer support vectors and overcomes other classical and orthogonal kernels.

1. Introduction

Support Vector Machines (SVMs) represent a machine learning model used to solve pattern recognition tasks such as classification, regression and outlier detection [1]. SVMs belong to a family of models for pattern analysis known as kernel-based learning methods that also include support vector clustering, kernel principal components analysis, kernel perceptron, etc. [2]; these methods are motivated by rigorous theoretical analysis and characterized by computational efficiency. Kernel methods enable practitioners to analyze nonlinear relations in data via the use of the so-called kernel trick. The first and arguably the most popular kernel method to date is the SVM. In the context of pattern classification, SVMs were originally developed to handle two-class problems [3,4]. Their solution involves a quadratic programming problem, with dual form [5]:

$$\begin{aligned} &\max\ L(\boldsymbol{\lambda}) = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \lambda_i \lambda_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t. } C \geq \lambda_i \geq 0\ \forall i = 1, \ldots, N, \text{ and } \sum_{i=1}^{N}\lambda_i y_i = 0 \end{aligned} \tag{1}$$

where C is a scalar parameter called the penalty factor, $\boldsymbol{\lambda}$ are non-negative Lagrange multipliers [4], and $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, represent a set of training data; $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th input vector and $y_i \in \{+1, -1\}$ its corresponding class label. Ultimately, those data points $\mathbf{x}_i$ for which the corresponding multipliers $\lambda_i \neq 0$ are the support vectors that define the decision boundary (a hyperplane in the feature space $\mathcal{H}$, which in general is a nonlinear boundary in the original space).

The function $K(\mathbf{x}, \mathbf{z})$ in (1), defined on $\mathbb{R}^d \times \mathbb{R}^d$ is called a kernel if there exists a mapping to the feature (Hilbert) space $\boldsymbol{\phi} : \mathbb{R}^d \to \mathcal{H}$ such that $K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$ [4]. The performance of SVMs is highly dependent on the kernel function, which can take different forms [6]. A variety of kernel functions have been developed for different problem domains, e.g. Linear [7]; String [8]; non-parametric [9]; etc. The difference between these kernels is in their own ability to represent the similarities between the data points in the feature space. Kernel optimization, kernel selection, and kernel design, have thus become important tasks in machine learning. In this paper, we address the problem of kernel design trough the study of orthogonal polynomial kernels, and introduce a new family of kernels based on Gegenbauer polynomials.

In the last two decades, orthogonal polynomials (OPs) have attracted interest as valid kernels for SVMs [6,10–14]. The most relevant studies are discussed in Section 2. The main idea behind building orthogonal polynomial kernels relies on the fact that the mapping $\boldsymbol{\phi}(x)$ for $x \in \mathbb{R}$ can be defined

* Corresponding author.
E-mail address: alfonso.rojas@gmail.com (A. Rojas-Domínguez).
[1] CONACYT research fellow.

https://doi.org/10.1016/j.patcog.2018.07.010
0031-3203/

in terms of n-th degree polynomials $\{p_n(x)\}_{n=0}^{\infty}$ as $\boldsymbol{\phi}(x) := (p_0(x),\ p_1(x),\ p_2(x), \ldots,\ p_n(x))$. Previous works on the generation of orthogonal polynomial kernels have shown that the number of support vectors in the SVM tends to be smaller when OPs are used as kernels than when classical kernel functions are used [13–15]. According to the theory developed by Vapnik in 1998 [6], the expectation of the probability of error for the optimal hyperplane constructed on the basis of training samples of size N (the expectation taken over both training and test data) is upper-bounded according to:

$$\mathbb{E}R\left(\lambda^{(N)}\right) \leq \frac{\mathbb{E}\ \min\left\{\mathcal{K}^{(N+1)}, \left(\frac{\mathcal{D}^{(N+1)}}{\rho^{(N+1)}}\right)^2\right\}}{N+1} \tag{2}$$

where $\mathcal{K}$ is the number of (essential) support vectors, $\mathcal{D}$ is the maximum norm among support vectors, ρ is the maximal possible margin between the separating hyperplane and its closest vector and the superscript notation indicates the size of the training dataset. Thus, it follows that selecting the solution with the smallest number of support vectors (and/or increasing the amount of training data) can improve the generalization of an SVM (*cf.* Chapter 10 of [6]).

The main hypothesis that has been put forward to explain the relatively fewer support vectors required when using OPs as kernels is that these help to decrease data redundancy in the feature space. Our own conjecture is that orthogonal polynomial kernels induce kernel matrices with fewer significant eigenvalues and thus, less support vectors. Recently, generalization measures have been proposed based on spectral analysis of the kernel matrix [16] that may support these hypotheses. The goal in designing the optimal kernel function is therefore to achieve high classification performance with the smallest possible number of support vectors [17]. Optimality shall thus be measured under two criteria: the classification accuracy and the Proportion of Support Vectors (PSV) used to define the decision boundary: $\text{PSV} = 100 \times \mathcal{K}/N$.

The main contributions of this paper are: (i) Previous proposals of orthogonal polynomial kernels are analyzed and positioned within a novel formulation that satisfies the Mercer conditions [18] while correcting some problems encountered within previous formulations. (ii) The hypothesis that better orthogonal polynomial kernels can be constructed with the so far unexplored members of the ultraspherical-polynomials is put forward and, to test this hypothesis, a new Gegenbauer kernel family is introduced. Only one extra control parameter is required to select a specific member of this family. (iii) A comprehensive experimental evaluation of Gegenbauer against classical and orthogonal polynomial kernels is carried out through a robust methodology that emphasizes the optimization of hyper-parameters, so that kernels are compared on the fairest ground possible. It is shown that the proposed Gegenbauer kernel overcomes its predecessors in terms of the considered indexes: Classification Accuracy and PSV.

The remaining sections of this paper are structured as follows: Section 2 provides a brief theoretical background of orthogonal polynomials and a review of previous orthogonal polynomial kernels. In Section 3, our novel formulation to produce orthogonal polynomial kernels for classification is described and the Gegenbauer kernel functions constructed under this formulation are presented. The experimental methodology is described in Section 4, and experimental results are reported and discussed in Section 5. Finally, conclusions to this work and further directions to extend this research are offered in Section 6.

2. Related work

2.1. Orthogonal polynomials

Orthogonal polynomials (OPs) can be regarded as universal function-approximators used in a variety of fields; their main quality is to reduce the redundancy in data, and they have succeeded in applications such as interpolation, data compression, quadrature approximation [19] and recently as pattern recognition techniques for feature extraction [20] and regression [21]. A complete survey on the theory supporting OPs can be consulted in [22]. However, the following definitions summarize the theory needed to explain the construction of orthogonal polynomial kernels.

Orthogonal Polynomial (OP). A polynomial $p(x)$ is said to be an orthogonal polynomial with respect to an inner product $\langle\cdot,\ \cdot\rangle$ if, given the space of n-th degree polynomials of one variable, Π_n, the inner product [23]:

$$\langle p(x), q(x)\rangle = 0\ \forall\ p(x) \in \Pi_n,\quad \text{degree}(q(x)) \neq \text{degree}(p(x)) \tag{3}$$

Orthogonal Polynomial Sequence (OPS). If the inner product $\langle p_m(x),\ p_n(x)\rangle$ is defined via a non-negative weight function $w(x)$ integrable on the open interval $(a,\ b)$, the sequence of n-th degree polynomials of one variable $\{p_n(x)\}_{n=0}^{\infty}$ is called an orthogonal polynomial sequence with respect to $w(x)$ on $(a,\ b)$ [24]; and the condition of orthogonality is given by:

$$\langle p_m(x),\ p_n(x)\rangle = \int_a^b p_m(x)\,p_n(x)\,w(x)\,dx = 0,\ \forall\ m \neq n \tag{4}$$

If instead of the continuous interval, (a, b), $w(x)$ is defined on a set of isolated points V in $\mathbb{R}$; then $\{p_n(x)\}_{n=0}^{\infty}$ is an OPS $\{p_n: 0 \leq n \leq |V|\}$, that satisfies [25]:

$$\langle p_m(x),\ p_n(x)\rangle = \sum_{x\ \in\ V} p_m(x)\,p_n(x)\,w(x) = 0,\ \forall\ m \neq n \tag{5}$$

The condition of orthogonality entails the property that any inner product between a pair of orthogonal polynomials of different degrees becomes zero. The condition is satisfied by polynomials from every known family of orthogonal polynomials, such as: Legendre, Chebyshev, Gegenbauer (ultraspherical), Laguerre, Hermite and Jacobi (characteristics of all of these polynomials can be consulted in the Askey scheme, *cf.* chapter 18 of [26]).

The property just described can be directly employed in the formulation of kernels based on orthogonal polynomials by including only products between polynomials of the same degree in the kernel design; these kernels are called Orthogonal Polynomial Kernels. Previous proposals for construction of this type of kernels are reviewed in the next subsection.

2.2. Previous orthogonal polynomial kernels

In 1998, Vapnik [6] described a number of kernels, including kernels that generate expansion of polynomials; in particular, he described a kernel based on Hermite polynomials $H(x)$ as follows:

$$K_H(x, z) = \sum_{i=0}^{\infty} q^i H_i(x) H_i(z) \tag{6}$$

where $0 \leq q \leq 1$ is a convergence factor. Vapnik [6] also demonstrated that, for multidimensional basis functions that are tensor products of the coordinatewise basis, a d-dimensional kernel is obtained by the product of d kernels of the form in (6):

$$K(\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{d} K_j(x_j, z_j) \tag{7}$$

In 2006, Ye et al. reported some of the first practical experiments with orthogonal polynomial kernels, and proposed a Chebyshev kernel in terms of univariate polynomials as [10]:

$$K_{Cheby}(x,z) = \frac{\sum_{i=0}^{n} T_i(x)T_i(z)}{\sqrt{1-xz}} \tag{8}$$

where $(\sqrt{1-xz})^{-1}$ corresponds to the weight function $w(x, z)$ and $T_i(x)$ is the i-th degree Chebyshev type-I polynomial. In order to compute the kernel function from input vectors $\mathbf{x}$ and $\mathbf{z}$ of dimension d, the following equation is used [13]:

$$K_{Cheby}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d} \frac{\sum_{i=0}^{n} T_i(x_j)T_i(z_j)}{\sqrt{1-x_j z_j}} \tag{9}$$

Similarly, other orthogonal polynomial kernels can be produced by substituting each $T_i(x_j)$ by the desired polynomials and an appropriate weight function. For instance, the Legendre kernel can be formulated using the Legendre polynomials $P_i(x_j)$ with $w(x_j, z_j) = 1$ as [11]:

$$K_{Legen}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d} \sum_{i=0}^{n} P_i(x_j)P_i(z_j) \tag{10}$$

In 2008, Ozer and Chen [12] extended the work of Ye et al. by expressing the kernel function in vector form originally, instead of doing it for each scalar, and defined the Generalized Chebyshev Kernel (GCK) and the Modified Chebyshev Kernel (MCK):

$$K_{GCK}(\mathbf{x},\mathbf{z}) = \frac{\sum_{i=0}^{n} T_i(\mathbf{x})T_i^{\mathrm{T}}(\mathbf{z})}{\sqrt{d-\langle \mathbf{x},\mathbf{z}\rangle}} \tag{11}$$

$$K_{MCK}(\mathbf{x},\mathbf{z}) = \frac{\sum_{i=0}^{n} T_i(\mathbf{x})T_i^{\mathrm{T}}(\mathbf{z})}{\exp(-\gamma\,||\mathbf{x}-\mathbf{z}||^2)} \tag{12}$$

where $T_i(\mathbf{x})$ is a row vector. In 2011, Ozer et al. experimented with GCK and MCK on few instances of pattern classification problems [13]. A more comprehensive set of experiments on pattern classification using SVMs with RBF kernel and GCK (among other classifiers) were carried out recently by Sun et al. [15]. In spite of the fact that the PSV was not used to measure the performance of the SVMs, the classification accuracies reported on twenty one test datasets provide an important reference for our experimental comparison.

In 2016, Moghaddam and Hamidzadeh showed that the use of modified Hermite ($He(x)$) polynomials in the formulation of a kernel function does indeed lead to a decrease in the PSV [14]. They defined the Hermite-based kernel as:

$$K_{Herm}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d} \sum_{i=0}^{n} He_i(x_j)He_i(z_j) \tag{13}$$

Moghaddam and Hamidzadeh evaluated their proposed kernel on fifteen real-world datasets and conclude that it also improves classification accuracy and training time [14].

The orthogonal polynomial kernels discussed in this work are listed in Table 1. The *Annihilation* and *Explosion* effects referred to in this table are described in later sections. A more detailed analysis of each individual kernel can be found in the corresponding references provided. In this work, our aim is to perform a robust and uniform evaluation of these kernels (with the exception of the one proposed by Vapnik for which there are no experimental results reported in the literature) and compare them against our own proposal.

Our proposal follows a construction method similar to the formulation first described by Vapnik and used in Eqs. (7), (9), (10) and (13). The difference between previous proposals and our formulation lies in the fact that we include a weight function of two variables and a scaling function designed to eliminate undesired effects that may hinder the operation of the orthogonal polynomial kernels. This is described in detail in the next section, where new kernels based on the Gegenbauer family of polynomials are developed following our novel formulation.

3. A novel formulation of orthogonal polynomial kernels

This section describes our proposal of a new formulation of orthogonal polynomial SVM kernels. In this formulation, an SVM kernel is constructed as the tensor product of the inner product of weighted and scaled univariate polynomials, i.e. kernel functions of scalar inputs which are then aggregated (under Mercer closures) to produce a vector-form kernel. Denoting the weight function by $w(x_j, z_j)$ and the scaling function by $u(p_i)$, the general form of an orthogonal polynomial SVM kernel under our formulation is thus given by:

$$K(\mathbf{x},\mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z})\rangle = \prod_{j=1}^{d} \sum_{i=0}^{n} p_i(x_j)\,p_i(z_j)w(x_j,z_j)u(p_i)^2 \tag{14}$$

The weight function is independent of the degree of the polynomials; this function is not unique, but it is selected from known functions to satisfy the orthogonality condition for each particular family of polynomials. The scaling function, $u(p_i) \equiv \beta\,|p_i(\cdot)|_{max}^{-1} \in \mathbb{R}^+$, is the reciprocal of the maximum absolute value of an i-th degree polynomial within its operational range, and β is a convenient positive scalar factor (see Section 3.1.2). Eq. (14) is general in the sense that it allows the creation of new kernels and it encompasses most previous proposals.

It has been observed [10,13,21] that a kernel combination of the form $\prod_{j=1}^{d} K(x_j, z_j)$ will yield a very small value if even a single one $K(x_j, z_j)$ is close to zero. Hereinafter this undesired effect is referred to as the *Annihilation Effect*. In trying to overcome this problem, some researchers in previous works [10,13,21] have empirically modified the weight function in the denominator of each $K(x_j, z_j)$ by adding a small constant $\varepsilon \ll 1$. For instance, the denominator $\sqrt{1-x_j z_j}$ in (9) becomes $\sqrt{1-x_j z_j+\varepsilon}$ in experiments [27]. We propose a different way to deal with the annihilation effect. Recall that for every family of orthogonal polynomials, $p_0(x) = 1$; thus, it immediately follows that the premise $K(x_j, z_j) \to 0$ can only occur if $\sum_{i=1}^{n} p_i(x) \to -1$. The proposed Eq. (14) includes a scaling function $u(p_i)$ that modifies the series $\sum_{i=1}^{n} p_i(x)$ such that each $|p_i(x)| \le 1$. Details will be provided in Section 3.1.2.

A second undesired effect occurs whenever $K(x_j, z_j) \gg 1$ and/or the dimensionality d of the problem is relatively big. In this case the total output of $|\prod_{j=1}^{d} K(x_j, z_j)| \to \infty$. This effect, deemed the *Explosion Effect*, will lead to numerical problems in the implementation of the kernels. The proposed scaling function in (14) is designed to avoid this problem. Details and proofs are contained in the sections below.

In the next section a new family of orthogonal polynomials is presented. The use of this family in the construction of orthogonal polynomial kernels poses a number of advantages, such as: increased capability to deal with high-dimensional problems, analytical tractability, and inclusion of other kernels.

3.1. The Gegenbauer kernel functions

A new family of kernels that follows our proposed formulation established in Eq. (14) is presented. This family is based on Gegenbauer polynomials, which includes both Chebyshev and Legendre kernels. The following discussion describes the family of polynomials as well as its corresponding weight and scaling functions.

Table 1
List of the orthogonal kernels discussed in this work.

Kernel (main ref.)	Short Label	Observations
Hermite [6]	K_H	First OP kernel described. Involves an infinite series with a convergence factor. No experimental results have been found on this kernel.
Chebyshev [10]	K_{Cheby}	Reports the first experimental results with OP kernels for SVM classifiers. Suffers from the *Annihilation* and *Explosion* effects.
Generalized Chebyshev [13]	K_{GCK}	First OP kernel formulated directly in vector form. Improves K_{Cheby} and defines one way to avoid the *Annihilation* effect.
Modified Chebyshev [13]	K_{MCK}	Kernel formulated in vector form (including an exponential function). Has shown better performance on non-linear problems than K_{GCK}.
Legendre [11]	K_{Legen}	Its advantage is that it does not require a weight function. It is free from the *Explosion* effect, but suffers from the *Annihilation* effect.
Hermite [14]	K_{Herm}	Prone to the *Annihilation* and *Explosion* effects. Hermite polynomials are orthogonal in $(-\infty, \infty)$, requiring manipulation of the input data. No weight function was originally defined for this kernel.
This Work, Eq. (20)	K_{Gegen}	Constructed according to Eq. (14). Avoids the *Annihilation* and *Explosion* effects. Includes and improves K_{Cheby} and K_{Legen} as special cases.
This Work, Eq. (30)	K_{s-Herm}	Constructed according to Eq. (14). Avoids the *Annihilation* and *Explosion* effects. Corrects problems of the K_{Herm} kernel.

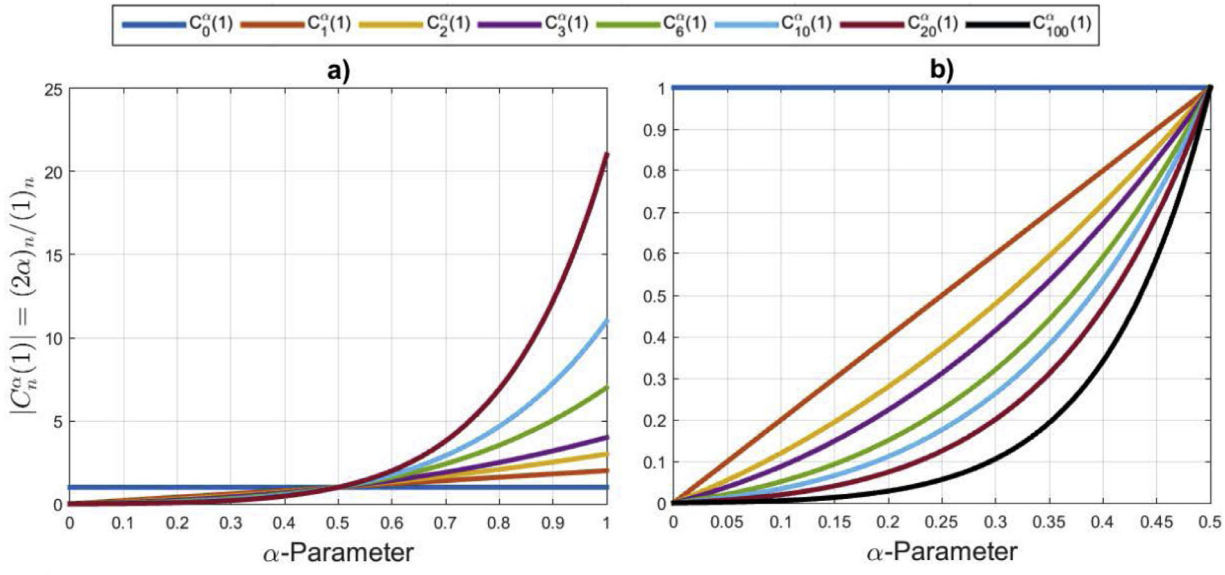

Fig. 1. Maximum values obtained by the scaling function for the Gegenbauer polynomials. a) Case of $\alpha \in (0, 1]$; b) case of $\alpha \in (0, 0.5]$ where the amplitude grows with α but decreases with the degree.

3.1.1. Gegenbauer polynomials

The Gegenbauer polynomials of degree n and parameter α, denoted as $C_n^\alpha(x)$, are solutions of the Gegenbauer differential equation [28]: $(1-x^2)y'' - (2\alpha+1)xy' + n(n+2\alpha)y = 0$ and produced by means of the three-term recurrence equation, Eq. (15) with $C_0^\alpha(x) = 1$, $C_1^\alpha(x) = \alpha 2x$. An infinite number of ultraspherical polynomials can be derived from the Gegenbauer family, one for each possible value of $\alpha > -0.5$. These polynomials along with an appropriate weight function $w_\alpha(x)$ satisfy (4) for $-1 \leq x \leq 1$; i.e., the generated polynomials are orthogonal in this range of x with respect to $w_\alpha(x)$ [29]. By substituting $\alpha = 0.5$, and $\alpha = 1$ in Eq. (15), the Legendre and Chebyshev type-II polynomials are obtained, respectively. Any other value of $\alpha > -0.5$ can be used, with the exception of $\alpha = 0$ (Chebyshev type-I), in whose case Eq. (16) should be used. Fig. 1 shows the maximum amplitudes of the Gegenbauer polynomials as function of α.

$$(n+1)C_{n+1}^\alpha(x) = 2(n+\alpha)xC_n^\alpha(x) - (n+2\alpha-1)C_{n-1}^\alpha(x) \tag{15}$$

$$C_{n+1}^{\alpha=0}(x) = 2xC_n^{\alpha=0}(x) - C_{n-1}^{\alpha=0}(x); \text{ with } C_0^{\alpha=0}(x) = 1,\ C_1^{\alpha=0}(x) = x \tag{16}$$

3.1.2. The weight and scaling functions

The Gegenbauer polynomials can be classified into two groups according to the parameter α: the first group for $-0.5 < \alpha \leq 0.5$ (including two special cases: Chebyshev type-I for $\alpha = 0$ and Legendre for $\alpha = 0.5$), and the second group with $\alpha > 0.5$. Regarding the first group, observe in Fig. 2 that their amplitude $|C_n^\alpha(1)| \leq 1$, such that a scaling function is not required for polynomials in this group. Similarly, the weight function for this group equals one.

In the second group, the growth of the polynomials amplitude to values $\gg 1$ is evident (Fig. 3a illustrates two cases), which can lead to the Explosion Effect. A mechanism to avoid this undesired effect is given in the form of the weight and scaling functions. The weight function is fixed for each family of orthogonal polynomials; in the case of Gegenbauer this is [23,29]:

$$w_\alpha(x) = (1-x^2)^{\alpha-1/2} \tag{17}$$

This function was used to produce the polynomials in Fig. 3b. Still, for $\alpha > 1$ it can be observed that the magnitude of the polynomials $\gg 1$. Thus, to completely eliminate the Explosion and Annihilation effects, the Gegenbauer polynomials in the second group need to not only be weighted but also scaled (Fig. 3c). According to our formulation, the scaling function is $u(p_i) \equiv \beta |p_i(\cdot)|_{max}^{-1} \in \mathbb{R}^+$. The Gegenbauer polynomials possess the property that their maximum amplitudes are always reached for $x = \pm 1$. Consequently, the

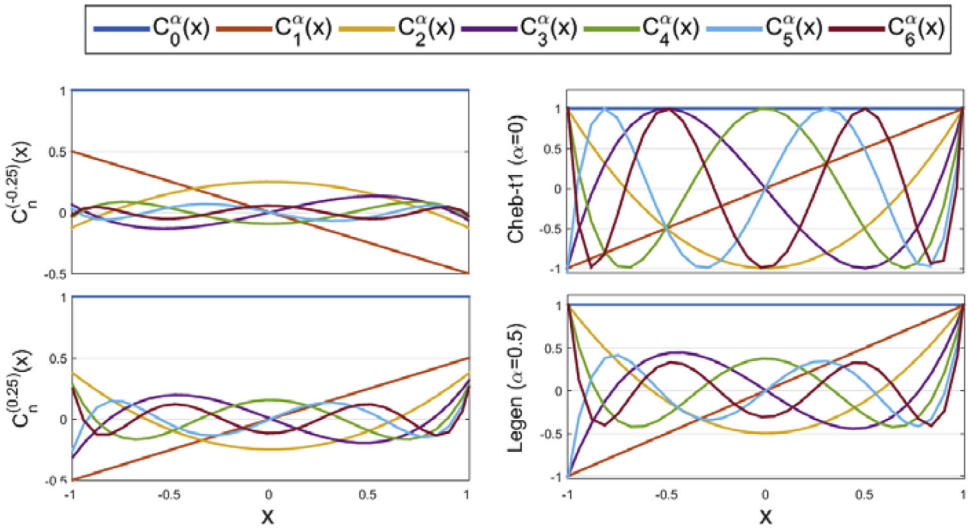

Fig. 2. Examples of Gegenbauer polynomials in the first group.
These polynomials do not require a scaling function since their amplitude $|C_n^\alpha(1)| \leq 1$.

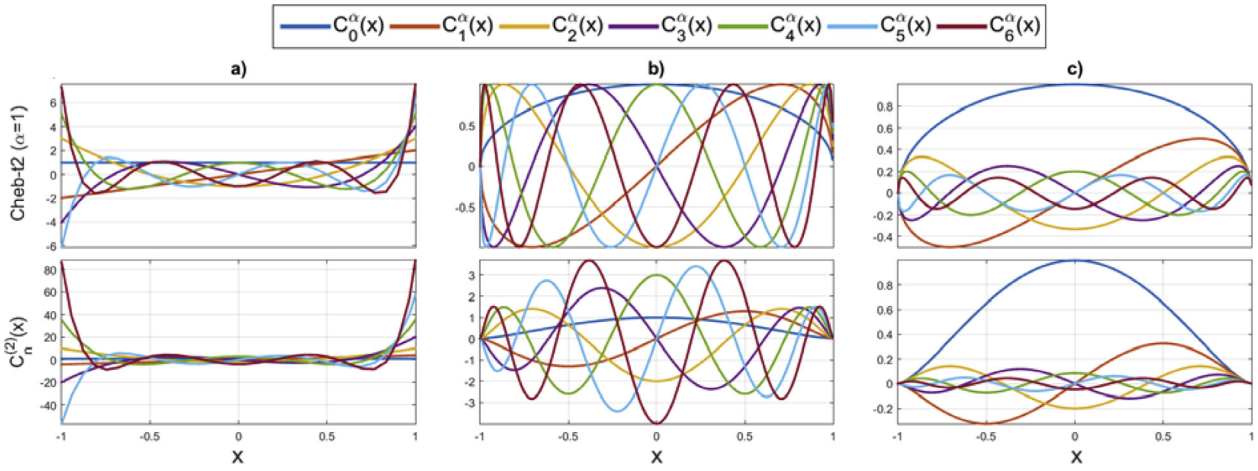

Fig. 3. a) Original, b) weighted, c) weighted-and-scaled Gegenbauer polynomials in the second group. These polynomials require weighting and scaling to eliminate the Annihilation and Explosion effects.

scaling function can be computed by means of the Pochhammer operator $(a \in \mathbb{R})_{n\in\mathbb{Z}} = a(a+1)(a+2)\cdots(a+n-1)$, given $\alpha > 0.5$ and $i > 0$ as: $|C_i^\alpha(1)| = (2\alpha)_i/(1)_i$. Setting $\beta = 1/\sqrt{n+1}$, the scaling function becomes:

$$u(p_i) = u\left(C_i^\alpha\right) = \left(\sqrt{n+1}\ \left|C_i^\alpha(1)\right|\right)^{-1} \tag{18}$$

3.1.3. Weight function of two variables

The analysis presented above is valid for univariate polynomials; however, although the shape of the weight function is preserved and independent of input data, when using Gegenbauer polynomials as kernels, two variables must be considered. Following the recent theory developed by Dunkl and Xu [23], the bivariate weight function is defined by taking the product of the corresponding univariate functions:

$$w_\alpha(x,z) = ((1-x^2)(1-z^2))^{\alpha-1/2} + \ \varepsilon \tag{19}$$

For the first group of Gegenbauer polynomials ($-0.5 < \alpha \leq 0.5$), the weight function is approximately $= 1$ for every point except nearby the extremes, as shown in Fig. 4a. Further, the amplitude of these polynomials is already small enough not to require the scaling by a weight function. Consequently, for this group of polynomials a convenient approximation is to use $w_\alpha(x,z) = 1$ which substantially simplifies the computation of the corresponding kernels.

For the second group of polynomials ($\alpha > 0.5$), the weight function becomes zero at values on the border (Fig. 4b), which causes problems, particularly if a dataset contains categorical features (because many instances could be mapped to these extreme values causing data loss). In this case, adding a small offset (e.g. $\varepsilon = 0.1$) solves the problem.

Up to this point, we have described all the elements required to formulate the Gegenbauer kernel functions. Now we can define the Gegenbauer kernel as:

$$K_{Geg}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d}\sum_{i=0}^{n} C_i^\alpha(x_j)C_i^\alpha(z_j)w_\alpha(x_j,z_j)u\left(C_i^\alpha\right)^2 \tag{20}$$

3.2. Theoretical properties

A valid kernel function must satisfy the necessary and sufficient conditions established in the Mercer's theorem [18]. Briefly stated, this theorem affirms that, given a finite input space $\mathbf{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\}$, the series $\sum_{i=1}^{\infty} \nu_i\theta_i(\mathbf{x})\theta_i(\mathbf{z})$, in terms of eigenfunctions $\theta_i \in L_2(\mathbf{X})$ and positive associated eigenvalues ν_i, converges absolutely and uniformly to $K(\mathbf{x}, \mathbf{z})$ when the latter is symmetric

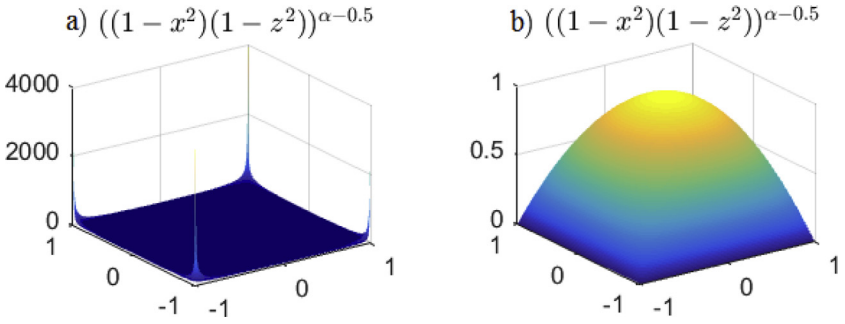

Fig. 4. Weight functions for the a) first group and b) second group of Gegenbauer kernels.

and positive semidefinite. Combinations of these kernels such as: summation or multiplication between two kernels; multiplication of a kernel with a positive real number; or composition of two kernels, produce other valid kernels because these combinations are defined as closures. For a more formal statement of this theorem and proofs of the closures, *cf.* Section 3.3 in [30]. Below, three important theoretical properties of the Gegenbauer kernel are described.

3.2.1. The Gegenbauer kernel is a valid Mercer kernel

A valid Mercer kernel is symmetric and positive semidefinite (non-negative definite). The proof that (20) is symmetric is trivial. We shall focus on demonstrating that it is positive semidefinite. According to the Mercer's theorem [18]: given a symmetric and continuous function $K(x, z)$ defined in the closed square $a \le x \le b$, $a \le z \le b$; and any function $g(\cdot)$ that ranges the class of all functions which are continuous in the closed interval $[a, b]$, the sufficient condition for $K(x, z)$ to be positive is:

$$\int_a^b \int_a^b K(x,z)g(x)g(z)dxdz \ge 0 \tag{21}$$

Theorem 1. *The function given in Eq. (20) is a positive Mercer kernel.*

Proof. first notice that $K_{Geg}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d} K_{Geg}(x_j, z_j)$, so that the Gegenbauer kernel defined for scalar data x and z is expressed as:

$$K_{Geg}(x,z) = \sum_{i=0}^{n} C_i^{\alpha}(x)C_i^{\alpha}(z)w_{\alpha}(x,z)u\left(C_i^{\alpha}\right)^2 \tag{22}$$

Substitution of (22) into (21) leads to (the integral limits are ignored for compactness):

$$\iint K_{Geg}(x,z)g(x)g(z)dxdz$$
$$= \iint \sum_{i=0}^{n} C_i^{\alpha}(x)C_i^{\alpha}(z)w_{\alpha}(x,z)u\left(C_i^{\alpha}\right)^2 g(x)g(z)dxdz$$

substituting (19), which is separable:

$$= \iint \sum_{i=0}^{n} C_i^{\alpha}(x)C_i^{\alpha}(z)\left[(1-x^2)^{\alpha-\frac{1}{2}}(1-z^2)^{\alpha-\frac{1}{2}} + \varepsilon\right]$$
$$\times u\left(C_i^{\alpha}\right)^2 g(x)g(z)dxdz$$

by definition, $u(C_i^{\alpha})$ is always positive and independent of the data, hence:

$$= \sum_{i=0}^{n} u\left(C_i^{\alpha}\right)^2 \iint C_i^{\alpha}(x)C_i^{\alpha}(z)(1-x^2)^{\alpha-1/2}(1-z^2)^{\alpha-1/2}g(x)g(z)dxdz$$
$$+ \sum_{i=0}^{n} \varepsilon u\left(C_i^{\alpha}\right)^2 \iint C_i^{\alpha}(x)C_i^{\alpha}(z)g(x)g(z)dxdz$$
$$= \sum_{i=0}^{n} u\left(C_i^{\alpha}\right)^2 \int C_i^{\alpha}(x)(1-x^2)^{\alpha-\frac{1}{2}}g(x)dx \int C_i^{\alpha}(z)(1-z^2)^{\alpha-\frac{1}{2}}g(z)dz$$
$$+ \sum_{i=0}^{n} \varepsilon u\left(C_i^{\alpha}\right)^2 \int C_i^{\alpha}(x)g(x)dx \int C_i^{\alpha}(z)g(z)dz$$
$$= \sum_{i=0}^{n} u\left(C_i^{\alpha}\right)^2 \left(\int C_i^{\alpha}(x)(1-x^2)^{\alpha-\frac{1}{2}}g(x)dx\right)^2$$
$$+ \sum_{i=0}^{n} \varepsilon u\left(C_i^{\alpha}\right)^2 \left(\int C_i^{\alpha}(x)g(x)dx\right)^2 \ge 0 \tag{23}$$

Thus, $K_{Geg}(x, z)$ is positive semidefinite, and it has been proved that it is a valid Mercer kernel.

Secondly, as the product of two kernels is also a kernel [1], $K_{Geg}(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d} K_{Geg}(x_j, z_j)$ for vector data $\mathbf{x}$ and $\mathbf{z}$ is also a valid Mercer kernel under this closure. ■

3.2.2. The Gegenbauer kernel prevents the adverse annihilation effect

The annihilation effect occurs when the summation in $K(\mathbf{x},\mathbf{z}) = \prod_{j=1}^{d}\sum_{i=0}^{n} p_i(x_j)p_i(z_j)$ produces 0 for at least one of the d dimensions. In the case of the Gegenbauer kernel this effect can only occur if $\sum_{i=1}^{n} C_i^{\alpha}(x_j)C_i^{\alpha}(z_j)w_{\alpha}(x_j,z_j)u(C_i^{\alpha})^2 = -1$, since $C_0^{\alpha}(x)C_0^{\alpha}(z)w_{\alpha}(x,z)|C_0^{\alpha}(1)|^{-2} = 1$. Therefore, to prove that this effect is avoided, it must be proved that the summation for degrees $i = 1 : n$ cannot be equal to -1.

Proof. for any $-1 < x < 1$, $-1 < z < 1$, $n \ge 1$, and $\alpha > -0.5$:

$$\left|\sum_{i=1}^{n} C_i^{\alpha}(x)C_i^{\alpha}(z)w_{\alpha}(x,z)u\left(C_i^{\alpha}\right)^2\right|$$

which by substitution of (18):

$$\le \sum_{i=1}^{n} \left|C_i^{\alpha}(x)\right|\left|C_i^{\alpha}(z)\right||w_{\alpha}(x,z)|\left(\left|C_i^{\alpha}(1)\right|\sqrt{n+1}\right)^{-2}$$

the maximum value of $|C_i^{\alpha}(\cdot)|=|C_i^{\alpha}(1)|$, thus:

$$\le \sum_{i=1}^{n} \left|C_i^{\alpha}(1)\right|^2|w_{\alpha}(x,z)|\left(\left|C_i^{\alpha}(1)\right|\sqrt{n+1}\right)^{-2} \tag{24}$$

$$\le \sum_{i=1}^{n} \frac{|w_{\alpha}(x,z)|}{n+1}$$

the maximum of $|w_{\alpha}(x_j, z_j)| = 1$, therefore:

$$\le \sum_{i=1}^{n} \frac{1}{n+1} < 1$$

Since the absolute value of the summation is always strictly less than 1, the summation can never be equal to -1. Further, for the

extreme values $(x, z) \in \{(1,1),\ (1,-1),(-1,1),(-1,-1)\}$

$$\sum_{i=0}^{n} C_i^{\alpha}(x_j) C_i^{\alpha}(z_j) w_{\alpha}(x_j, z_j) u\left(C_i^{\alpha}\right)^2 = \sum_{i=0}^{n} (\pm 1) w_{\alpha}(x_j, z_j) = \sum_{i=0}^{n} \pm\varepsilon \leq (n+1)\varepsilon \tag{25}$$

which can be guaranteed to be ≤ 1 by requiring that $\varepsilon \leq 1/(n+1)$. ■

3.2.3. The Gegenbauer kernel prevents the adverse explosion effect

The explosion effect occurs when the product of polynomials $C_n^{\alpha}(x) C_n^{\alpha}(z)$ powered by the problem dimension d produces large values that can lead to numerical issues. To prove that this effect is prevented, it must be proved that $K_{Geg}(\mathbf{x}, \mathbf{z})$ is upper-bounded.

Proof. for any $-1 \leq x \leq 1$, $-1 \leq z \leq 1$, $n \geq 0$, $d \geq 1$ and $\alpha > -0.5$:

$$K_{Geg}(\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{d} \sum_{i=0}^{n} C_i^{\alpha}(x_j) C_i^{\alpha}(z_j) w_{\alpha}(x_j, z_j) \left(\left| C_i^{\alpha}(1) \right| \sqrt{n+1} \right)^{-2}$$

following similar steps to those in (24):

$$K_{Geg}(\mathbf{x}, \mathbf{z}) \leq \prod_{j=1}^{d} \left(\sum_{i=0}^{n} \frac{1}{n+1} \right) \leq \prod_{j=1}^{d} \left(\frac{n+1}{n+1} \right) \leq 1 \tag{26}$$

Thus, it has been proved that the Gegenbauer kernel, $K_{Geg}(\mathbf{x}, \mathbf{z}) \leq 1$. ■

4. Experimental methodology

In this section, our aim is to compare the Gegenbauer kernel against classical kernels (Eqs. (27) to (29)) and previously described orthogonal polynomial kernels on the fairest ground possible, by means of individual hyper-parameter optimization and evaluation on a variety of datasets. The Hermite kernel formulation in Eq. (13) is very prone to the Explosion Effect, and was adapted under our formulation to the form in Eq. (30).

$$K_{Linear}(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathrm{T}}\mathbf{z} \tag{27}$$

$$K_{RBF}(\mathbf{x}, \mathbf{z}) = e^{-\gamma ||\mathbf{x}-\mathbf{z}||^2} \tag{28}$$

$$K_{Polynomial}(\mathbf{x}, \mathbf{z}) = \left(a\mathbf{x}^{\mathrm{T}}\mathbf{z} + b\right)^n \tag{29}$$

$$K_{s-Herm}(\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{d} \sum_{i=0}^{n} He_i(x_j) He_i(z_j) \left(e^{-\left(x_j^2 + z_j^2\right)/2} \right) \left(2^{-2n}\right) \tag{30}$$

Fig. 5 shows the general process employed to evaluate seven state-of-the-art kernels (RBF, MCK, LINEAR, GCK, POLY, LEGEN and CHEB) against the two kernels following our formulation (GEGEN and s-HERM). Hereafter, these short labels will be used for the sake of presentation. The evaluation was carried out on 20 datasets for binary classification taken from the UCI Machine Learning,[1] LIBSVM,[2] and KEEL[3] repositories. These datasets, described in Table 2, were scaled to $[-1, 1]$ in order to meet the requirements of orthogonal polynomial kernels and to avoid bias towards attributes of larger values.

Each of the nine kernels under evaluation has at least one hyper-parameter that needs to be tuned by the user to get the best performance on a specific dataset. We performed an automatic hyper-parameter optimization by means of an efficient metaheuristic called BUMDA [31]. A hyper-parameter vector $\boldsymbol{h} = (C, \gamma, a, n, \alpha)$ constitutes and individual of the population $\boldsymbol{H}$ used by this metaheuristic. The specification of an initial population size (100 in this work) is the only parameter required by BUMDA at the beginning of the optimization. The ranges in which the best hyper-parameters were searched are the following: $C \in (0, 2^5]$, $n \in \{1, 2, \ldots, 6\}$, $\gamma \in (2^{-6}, 2^2]$. The offset factor of the classical polynomial kernel was set to $b = 0$ and its scale factor a was searched in the same range as γ. For the Gegenbauer kernel, the parameter α was searched in $(-0.5, 1.5]$.

The quality of each kernel using a particular hyper-parameter setting was measured by the classification accuracy obtained by training an SVM with the LIBSVM solver [38] and evaluating it by 10-fold cross-validation. The same data folds and initial population $\boldsymbol{H}^0$ of hyper-parameters were used by all the kernels to make the comparison between them as fair as possible. At the end of 15 generations, i.e. after evaluating around 1,500 different SVMs for each kernel, the optimization process ends by providing the best SVM kernel with its corresponding individual $\boldsymbol{h}^*$ and performance measures (accuracy, PSV and training time). All the performance measures and all the evaluated SVM kernels are stored for subsequent analysis. To increase the confidence in our results, the optimization process was repeated 35 times for each dataset (35 experimental trials). Each experimental trial utilizes a different 10-fold data partition and a different $\boldsymbol{H}^0$. The average performance measures of these 35 optimal kernels are finally reported and were used for statistical tests. The Friedman, aligned Friedman and Quade tests were employed in order to draw stronger conclusions from the statistical point of view. The difference between these three methods is in the way of ranking each algorithm; a complete description of these tests is found in [39,40].

A suggested implementation of the Gegenbauer kernel functions for SVMs is provided in Algorithm 1. The implementation of the Pochhammer operator used in (18) is provided in Algorithm 2. Regarding the determination of parameters α and n, any $\alpha > -0.5$, and $n \in \mathbb{Z}^+$ can be used. However, in practice some extra considerations have to be taken into account. The orthogonal polynomial kernels are implemented by recurrence equations and the time required for their evaluation increases exponentially with the degree. In addition, in our experiments and in previous works, setting $n \leq 6$ has sufficed to obtain good performance in a large number of prob-

Table 2
Datasets description and its classification accuracy indexes.

	Dataset(source)	#Instances	#Attributes	Reported accuracy (%) with RBF	[Ref.]
1	a1a[2]	1605	123	81.57	[32]
2	Australian[2]	690	14	81.87	[32]
3	Breast[1]	683	10	97.31	[33]
4	Diabetes[1]	768	8	77.73	[33]
5	Fourclass[2]	862	2	99.81	[33]
6	German[2]	1000	24	73.38	[32]
7	Glass2[3]	214	9	92.06	[34]
8	Haberman[1]	306	3	73.55	[33]
9	Heart[2]	270	13	87.70	[15]
10	Ionosphere[1]	351	34	94.53	[15]
11	Liver[1]	345	7	72.45	[15]
12	Monks-1[1]	124	6	95.01	[15]
13	Monks-2[1]	169	6	76.58	[15]
14	Monks-3[1]	122	6	89.37	[15]
15	plrx[1]	182	12	71.46	[35]
16	Sonar[2]	208	60	87.05	[15]
17	Splice[2]	1000	60	88.33	[36]
18	Vehicle[2]	1243	22	82.24	[37]
19	wdbc[1]	569	30	97.00	[32]
20	wpbc[1]	194	33	80.09	[15]

[1] archive.ics.uci.edu/ml/datasets.html.
[2] www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.
[3] sci2s.ugr.es/keel/dataset.php?cod=121.

Experimental Methodology

INPUT: User-defined parameters (hyper parameter ranges, number of trials, etc.)
OUTPUT: Performance Information (Accuracy and PSV) to carry out statistical analysis.

```
FOR EACH experimental trial (i) DO                                   // i :1 to 35 in this work
    H_i^0 ← Generate a new initial population of hyper-parameters
    FOR EACH d ∈ Datasets DO
        Partition Dataset_d into DataFolds = {Training_{j,d}, Test_{j,d}}_{j=1}^{10}   //for 10-fold cross validation
        FOR EACH k ∈ Kernels DO
            H_i^* ← BUMDA(H_i^0, Kernel_k, LIBSVM, DataFolds)        // hyper parameter optimization
            [Accuracy_{i,k}, PSV_{i,k}] ← SVM(Kernel_k, H_i^*)       // store performance indexes
        END
    END
END
```

Fig. 5. Pseudocode of Experimental Methodology. Refer to the main text for explanation of this figure.

Algorithm 1 The Gegenbauer kernel functions $K_{Geg}(\mathbf{x}, \mathbf{z}, n, \alpha)$.

INPUTS: $\mathbf{x}, \mathbf{z} \in [-1,1]^d$, $\alpha > -0.5$, and $n \in \mathbb{Z}^+$
OUTPUTS: $K_{Geg}(\mathbf{x}, \mathbf{z}, n, \alpha) \in \mathbb{R}$

```
DEFINE: mult ← 1, xlen ← lengthOf(x), ylen ← lengthOf(z), i ← 0 and j ← 0;
IF (α == 0) RETURN C(x_i, k, 0);                                          Eq. (16)
WHILE (i < xlen AND j < ylen)
    IF ( indexOf(x_i) == indexOf(z_j))
        DEFINE: sum ← 1;
        FOR (k = 1:n ):
            sum ← sum + C(x_i, k, α) × C(z_j, k, α) × w(x_i, z_j, α) × u(k, α)^2;   Eqs. (15, 18, 19)
        END
        mult ← mult × sum; i ← i + 1; j ← j + 1;
    ELSE
        IF ( indexOf(x_i) > indexOf(z_j)) j ← j + 1;
        ELSE i ← i + 1;
        END
    END
END
RETURN mult;
```

$$w(x_i, z_j, \alpha) = \begin{cases} 1, & -0.5 < \alpha \leq 0.5 \\ (\,(1-x^2)\,(1-z^2)\,)^{\alpha-1/2} + (\varepsilon = 0.1), & \alpha > 0.5 \end{cases} \quad \text{From Eq. (19)}$$

$$u(k, \alpha) = \beta \times |C_k^{\alpha}(1)| = (1/\sqrt{n+1}\,) \times (2\alpha)_k/(1)_k \quad \text{From Eq. (18)}$$

Algorithm 2 Pochhammer Operator $(x)_n$.

INPUTS: $x \in \mathbb{R}$, $n \in \mathbb{Z}$;
OUTPUT: $(x \in \mathbb{R})_{n \in \mathbb{Z}} = (x)(x+1)\ldots(x+n-1)$

```
IF (n == 0) RETURN 1.0;
IF (n < 0 OR x == 0) RETURN 0.0;
DEFINE aux ← 1;
FOR (i = 1 : n − 1) : aux ← aux × (x + i); END
RETURN aux;
```

lems. Because of these reasons, we recommend to first test small degrees when using orthogonal polynomial kernels as done in our methodology. With respect to α, we have noticed a tendency towards more training time required with values of α larger than 2 and without a gain in performance. Thus, we recommend employing an initial setting of $-0.5 < \alpha \leq 1.5$ for the Gegenbauer kernel.

5. Experimental results and discussion

Tables 3 and 4 summarize the results of the 35 trials of the experiment described in the previous section. In addition to the mean and standard deviation of the 35 best models found for each kernel, these tables include a ranking among the nine kernels. Average ranks measure the general performance of kernels and they are also used later on to apply the statistical tests. The performance of kernels in terms of accuracy can be observed in Table 3. The Gegenbauer kernel performs better than the rest, with the RBF and s-Hermite ranked 2nd and 3rd, respectively.

The general performance of the kernels in terms of PSV, complementing the accuracy index, is presented in Table 4. The linear and polynomial kernels obtain better ranks than the RBF kernel. The kernels s-Hermite and GCK, are ranked 1st and 2nd in PSV,

Table 3
Statistics on kernel performance according to classification accuracy.

Kernel	RBF		MCK		LINEAR		GCK		POLY		LEGEN		CHEB		s-HERM		GEGEN	
Dataset	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D	Mean	St.D
a1a	83.246	0.7	80.977	0.5	83.552	0.3	83.485	0.3	83.503.5	0.3	80.268	0.3	75.469	0.2	83.503.5	0.2	**83.64**1	0.2
Australian	86.064	0.4	84.467	2.6	85.496	0.1	86.005	0.4	86.103	0.3	80.218	0.8	79.959	0.7	86.152	0.3	**86.51**1	0.4
Breast	97.134	0.3	96.258	0.2	96.966	0.2	97.095	0.3	97.153	0.2	96.267	0.3	95.379	0.3	**97.37**1	5.8	97.292	0.2
Diabetes	**78.04**1	0.3	77.733	0.4	77.457	0.4	77.624	0.4	77.605	0.4	77.268	0.5	70.239	1.2	77.942	1.7	77.526	0.3
Fourclass	**100**1.5	0.0	**100**1.5	0.0	77.099	0.7	99.246	0.1	81.477	0.1	99.705	0.1	99.953	0.1	81.168	2.1	99.874	0.1
German	75.365	2.3	71.837	0.7	76.873	0.4	74.836	2.6	76.864	0.5	71.338	0.5	70.159	0.2	**77.20**1	0.6	76.922	0.5
Glass 2	**92.65**1	0.4	92.482	0.4	92.147.5	0.0	92.353	0.3	92.334	0.3	92.205	0.3	89.499	0.9	92.147.5	0.4	92.196	0.3
Haberman	74.813	0.6	75.022	0.7	73.428	0.3	74.436	0.5	74.217	0.7	74.624	0.4	72.449	0.9	74.445	0.7	**75.09**1	0.6
Heart	84.393	0.5	81.437	2.0	84.214	0.9	82.026	1.4	83.945	0.6	77.568	0.8	72.729	0.9	85.262	2.3	**85.98**1	0.5
Ionosphere	**95.88**1	0.3	95.042	0.4	89.388	0.7	91.897	0.6	92.246	0.6	92.825	0.6	69.229	8.2	94.123	4.0	93.374	0.4
Liver	74.253	0.6	**74.85**1	0.7	69.629	0.9	74.712	0.6	72.807	0.6	73.086	0.7	71.478	0.8	73.895	2.0	74.134	0.7
Monks-1	82.564	1.6	87.752	2.3	69.208	2.8	74.696	2.4	72.237	1.9	76.335	1.8	68.999	1.3	85.753	2.0	**97.52**1	0.9
Monks-2	81.954	1.7	**94.55**1	3.5	62.159	0.0	77.755	2.0	71.778	1.6	85.033	1.6	73.197	1.8	73.586	6.5	85.642	1.5
Monks-3	91.883	0.8	91.474	1.1	81.447	1.6	88.315	1.1	80.128	2.0	84.036	2.1	73.429	1.9	93.422	7.1	**93.46**1	0.1
plrx	73.402	0.8	**73.67**1	0.9	71.469	0.0	72.434	0.3	71.977	0.5	71.538	0.3	72.853	0.9	72.425	0.5	72.316	0.4
Sonar	90.532	1.0	89.923	1.0	80.087	1.1	86.686	1.0	87.295	1.2	73.558	0.9	70.219	0.8	89.914	1.6	**92.25**1	1.0
Splice	87.964	5.0	89.083	0.5	80.647	0.8	87.225	0.5	82.946	0.5	56.928	0.4	51.839	1.3	92.262	0.4	**93.62**1	0.3
Vehicle	84.566	3.6	84.695	1.5	82.737	0.2	**85.51**1	0.3	85.073	0.3	77.648	0.5	75.099	0.6	85.054	0.7	85.242	0.4
wdbc	98.232	0.2	97.247	0.4	97.876	0.2	98.094	0.2	97.925	0.2	95.459	0.4	95.728	0.4	**98.35**1	0.2	98.153	0.3
wpbc	82.132	1.5	78.327	1.1	81.515	1.0	81.406	1.2	81.804	1.2	76.409	0.2	76.548	0.4	82.113	1.6	**82.56**1	1.0
Avg. Rank	3.075		4.025		6.725		4.85		5.375		6.8		8.15		3.5		**2.5**	

but the s-Hermite kernel is ranked 3rd in accuracy while GCK is 5th.

The training times required for each combination of a kernel and a dataset are reported in Table 5. These values can be analyzed by groups based on accuracy: the top-3 kernels are Gegenbauer, RBF and s-Hermite; the second group, with medium performance includes the MCK, GCK and Polynomial kernels; the bottom-3 kernels are Linear, Legendre and Chebyshev. It can be observed that the top-3 kernels, with an average total time required of 44.5 (s), outperform the second group which requires an average of 236.2 (s) and also the bottom-3 group with 174.3 (s).

5.1. Comparison between all kernels

Results on accuracy and PSV reported in Tables 3 and 4, respectively, were obtained by means of an optimization of the parameters corresponding to each kernel and the hyper-parameter C, which represents the penalty factor in SVMs. In general, it was found that the kernel parameters have a stronger influence on performance of the SVMs than C. In every case, all measures were taken to ensure that the evaluation was as independent as possible from the choice of the parameter values, so that the true potential of each kernel could be appreciated.

The Gegenbauer, RBF and s-Hermite kernels emerge as the best three out of the nine considered in terms of accuracy. To verify this observation and to get a closer view of their differences in performance, the statistical tests of Friedman, Aligned Friedman and Quade were applied. The results of these tests, performed under a significance level of 0.05, are reported in Tables 6–8.

The statistical tests performed look for a significant difference between at least two kernels. Any p-value < 0.05 indicates that there exist at least two kernels with different performance. The results in Table 6 show that, according to the Friedman, Aligned Friedman and Quade tests, the performance between some kernels differ significantly with respect to the accuracy index (p-values 5.29E-11, 0.023 and 1.8E-13). According to the Aligned Friedman test, there are also differences among the performance of the kernels with respect to the PSV index (p-value 0.02). After *post hoc* tests of Bonferroni-Dunn, Holm, Holland and Finner [39] using Gegenbauer and GCK as the control methods under accuracy and PSV indexes, respectively, it was found that Gegenbauer is statistically superior in accuracy to all kernels except for RBF and s-Hermite and that the GCK is significantly better in PSV than the Legendre kernel.

In order to obtain a deeper understanding of the relative behavior of the proposed kernel, we performed separate statistical tests comparing the Gegenbauer kernel against the classical kernels and against the other orthogonal polynomial kernels. The results of these comparisons are discussed below.

Table 4
Statistics on kernels performance according to proportion of support vectors.

Kernel	RBF		MCK		LINEAR		GCK		POLY		LEGEN		CHEBY		s-HERM		GEGEN	
Dataset	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv	Mean	St.Dv
a1a	42.97	7.21	61.78	10.1	36.62	1.35	39.414	1.44	36.83	1.5	76.39	0.09	**3.10**1	2.27	41.06	0.79	40.25	1.25
Australian	38.46	9.16	39.67	11.6	**31.3**1	0.55	35.993	1.95	37.05	14.2	55.09	12.8	43.98	0.17	35.32	7.92	36.34	8.70
Breast	13.33	8.07	36.79	9.55	**8.37**1	1.32	9.382	2.36	14.04	11.7	26.67	4.54	28.28	3.70	14.265	5.80	14.46	5.44
Diabetes	55.46	3.17	56.49	4.87	54.104	2.88	55.637	3.94	55.648	4.35	51.72	1.96	**48.63**1	2.44	53.763	1.75	55.15	4.23
Fourclass	6.303.5	2.42	**4.17**1	0.92	55.179	5.84	9.015	1.78	38.47	0.85	6.303.5	1.06	4.882	0.51	51.708	2.17	10.56	0.90
German	63.96	12.9	84.47	8.7	**53.4**1	0.55	61.085	0.56	53.82	2.38	89.88	1.38	99.09	0.62	57.24	1.11	56.43	1.28
Glass2	29.68	7.69	25.16	7.62	27.107	0.67	20.612	1.59	24.45	3.45	**16.64**1	2.16	39.79	9.77	21.433	0.42	22.64	0.58
Haberman	54.46	0.59	53.82	0.45	54.325	0.27	54.787	0.42	54.88	0.99	**52.51**1	0.47	55.49	0.62	54.14	0.77	53.93	1.41
Heart	44.74	6.25	45.25	8.04	39.522	4.65	46.926	6.30	**37.76**1	2.45	66.19	2.19	62.98	0.18	43.013	2.34	47.57	3.78
Ionosphere	41.67	12.0	51.59	9.60	27.114	5.85	26.333	1.78	28.45	7.56	41.88	0.20	**21.43**1	3.98	24.922	4.06	32.76	2.91
Liver	71.37	3.95	66.03	4.06	75.799	2.07	68.314	4.18	68.55	3.73	65.82	4.46	**62.83**1	3.61	73.468	2.08	69.26	5.25
Monks-1	53.33	2.81	**33.7**1	6.50	74.219	10.1	56.165	13.15	62.26	7.54	66.87	3.47	70.88	1.62	53.694	2.02	47.22	3.82
Monks-2	73.88	11.2	**34.8**1	9.97	91.159	0.38	57.705	1.30	55.34	5.22	42.13	1.02	70.97	2.30	60.216	6.52	40.92	2.04
Monks-3	47.53	5.49	**38.5**1	9.52	52.74.5	4.81	58.218	6.97	54.87	9.10	52.74.5	9.96	62.79	3.24	44.562	7.16	53.96	8.78
plrx	94.97	4.28	96.28	0.76	70.283	0.94	70.554	0.86	72.25	1.89	**59.14**1	8.23	96.69	0.07	70.242	0.56	72.66	3.22
Sonar	67.77	8.35	60.55	5.95	60.074	10.7	**43.92**1	2.80	57.42	3.35	69.98	0.25	77.79	0.21	58.293	1.67	64.06	3.20
Splice	69.25	8.73	79.827	8.67	43.932	7.19	56.403	2.71	92.78	0.14	97.79	0.02	**1.99**1	1.72	71.536	0.45	61.84	3.35
Vehicle	42.08	8.18	39.114	1.90	41.557	0.24	39.103	0.70	**33.83**1	1.38	60.99	10.5	38.72	0.15	40.186	0.74	39.45	0.69
wdbc	12.49	1.76	11.26	4.49	11.58	0.22	11.417	1.67	10.644	2.86	10.95	3.57	10.42	0.08	**9.42**1	1.31	10.623	2.01
wpbc	57.97	9.57	81.378	14.0	**45.12**1	1.84	50.853	1.10	45.62	6.06	51.24	10.6	96.79	5.34	55.455	1.64	55.76	2.44
Avg. Rank	6.025		5.35		4.625		4.35		4.6		5.5		5.65		**4.15**		4.75	

Table 5
Training times (in seconds) of each kernel using its best parameters, Mean ± Std. Dev.

Dataset	RBF	MCK	LINEAR	GCK	POLY	LEGEN	CHEB	s-HERM	GEGEN
a1a	0.22 ± 0.1	38.91 ± 31.5	12.85 ± 5.1	6.82 ± 5.1	36.53 ± 25.3	1.13 ± 0.1	238.5 ± 28.9	7.33 ± 2.0	4.10 ± 3.9
Australian	0.04 ± 0.0	0.87 ± 3.6	3.57 ± 2.5	0.34 ± 0.3	0.26 ± 0.4	0.92 ± 0.5	0.41 ± 0.1	4.00 ± 2.4	2.88 ± 1.7
Breast	0.04 ± 0.1	6.48 ± 6.2	0.28 ± 0.3	0.17 ± 0.3	0.24 ± 0.3	0.32 ± 0.2	0.34 ± 0.3	0.40 ± 0.3	0.94 ± 1.6
Diabetes	0.26 ± 0.1	8.57 ± 33.1	0.28 ± 0.3	0.68 ± 1.0	5.75 ± 10.0	0.64 ± 0.3	2.65 ± 1.1	5.73 ± 3.6	5.21 ± 6.2
Fourclass	0.09 ± 0.2	1.94 ± 1.2	0.75 ± 0.6	3.05 ± 0.8	239.9 ± 192.0	0.66 ± 0.3	0.46 ± 0.4	4.38 ± 1.0	1.45 ± 0.4
German	0.28 ± 0.0	50.84 ± 44.9	7.85 ± 3.6	0.83 ± 0.7	4.14 ± 4.7	0.82 ± 0.1	1.40 ± 0.1	4.96 ± 3.9	4.90 ± 4.3
Glass2	0.02 ± 0.1	0.77 ± 1.5	0.003 ± 0.0	1.07 ± 1.3	0.34 ± 0.8	0.02 ± 0.0	0.36 ± 0.4	0.17 ± 0.2	0.22 ± 0.2
Haberman	0.09 ± 0.2	2.62 ± 14.5	0.07 ± 0.2	0.30 ± 0.4	40.19 ± 43.1	1.01 ± 1.0	0.38 ± 0.3	0.86 ± 0.5	1.08 ± 0.9
Heart	0.05 ± 0.1	1.81 ± 1.7	0.62 ± 0.4	1.27 ± 1.6	3.90 ± 2.2	0.31 ± 0.2	0.12 ± 0.2	1.30 ± 1.3	1.13 ± 0.6
Ionosphere	0.05 ± 0.1	0.57 ± 0.8	0.62 ± 0.7	0.22 ± 0.3	0.13 ± 0.2	0.71 ± 0.4	51.30 ± 22.3	2.55 ± 1.4	5.10 ± 3.9
Liver	0.09 ± 0.1	1.52 ± 10.0	0.29 ± 0.4	0.32 ± 0.6	3.74 ± 2.8	1.88 ± 1.1	0.57 ± 0.5	1.18 ± 1.9	0.78 ± 0.7
Monks-1	0.005 ± 0.0	0.79 ± 1.2	0.003 ± 0.0	0.12 ± 0.1	0.03 ± 0.1	0.01 ± 0.0	0.02 ± 0.0	0.003 ± 0.0	0.11 ± 0.0
Monks-2	0.11 ± 0.3	0.39 ± 1.0	0.04 ± 0.1	0.06 ± 0.1	0.92 ± 0.9	0.23 ± 0.3	0.02 ± 0.0	0.08 ± 0.2	0.26 ± 0.1
Monks-3	0.05 ± 0.2	0.21 ± 0.2	0.004 ± 0.0	0.02 ± 0.0	0.02 ± 0.1	0.05 ± 0.1	0.06 ± 0.1	0.14 ± 0.3	0.30 ± 0.2
plrx	0.15 ± 0.3	1.09 ± 1.1	1.22 ± 1.0	0.95 ± 0.9	0.05 ± 0.2	0.03 ± 0.1	0.57 ± 0.5	0.63 ± 0.5	0.72 ± 0.5
Sonar	0.09 ± 0.1	4.27 ± 3.9	0.10 ± 0.2	2.48 ± 2.7	0.11 ± 0.2	0.58 ± 0.5	0.60 ± 0.5	0.25 ± 0.3	3.63 ± 1.7
Splice	0.37 ± 0.1	13.89 ± 26.5	0.47 ± 0.3	9.78 ± 5.5	179.1 ± 131.4	3.44 ± 1.8	179.5 ± 55.9	17.55 ± 7.7	5.29 ± 3.8
Vehicle	0.34 ± 0.2	12.47 ± 85.6	0.83 ± 1.0	3.23 ± 4.2	4.82 ± 4.3	2.11 ± 1.1	0.01 ± 0.0	17.13 ± 8.1	21.42 ± 15.4
wdbc	0.02 ± 0.0	4.73 ± 4.1	0.03 ± 0.0	0.47 ± 0.3	0.03 ± 0.0	0.19 ± 0.1	0.08 ± 0.1	0.56 ± 0.6	1.25 ± 0.9
wpbc	0.06 ± 0.1	2.04 ± 2.9	0.15 ± 0.2	0.22 ± 0.2	1.11 ± 0.7	0.10 ± 0.2	0.21 ± 0.2	0.47 ± 0.5	0.68 ± 0.1
Sum of means	2.41	154.79	30.05	32.41	521.35	15.18	477.55	69.70	61.45

Table 6
Average ranks and p-values of statistical tests on all kernels.

Algorithm	Friedman		Aligned Friedman		Quade	
	Accuracy	PSV	Accuracy	PSV	Accuracy	PSV
RBF	3.075	6.025	55.775	104.425	3.176	5.876
MCK	4.025	5.350	73.775	103.200	3.619	5.424
LINEAR	6.725	4.625	116.475	84.075	6.855	4.283
GCK	4.850	4.350	82.100	**69.000**	5.110	**4.014**
POLYNOMIAL	5.375	4.600	94.175	81.650	5.698	4.338
LEGENDRE	6.800	5.500	123.000	107.950	6.548	6.117
CHEBYSHEV	8.150	5.650	148.950	107.550	8.224	5.605
s-HERMITE	3.500	**4.150**	69.800	80.150	3.576	4.600
GEGENBAUER	**2.500**	4.750	**50.450**	76.500	**2.195**	4.743
p-value	**8.16E-11**	0.35	**0.023**	**0.02**	**2.87E-14**	0.29

Table 7
Average ranks and p-values of tests on classical kernels vs Gegenbauer.

Algorithm	Friedman		Aligned Friedman		Quade	
	Accuracy	PSV	Accuracy	PSV	Accuracy	PSV
RBF	2.000	3.200	26.45	53.00	1.880	3.123
LINEAR	3.700	**2.100**	62.15	37.10	3.761	**2.262**
POLYNOMIAL	2.850	2.400	52.35	**35.80**	3.000	2.300
GEGENBAUER	**1.450**	2.300	**21.05**	36.10	**1.357**	2.314
p-value	**1.23E-07**	**0.038**	**0.0013**	**9.23E-04**	**9.30E-15**	**0.001**

Table 8
Average ranks and p-values of statistical tests on OP kernels vs Gegenbauer.

Algorithm	Friedman		Aligned Friedman		Quade	
	Accuracy	PSV	Accuracy	PSV	Accuracy	PSV
MCK	2.650	3.650	47.100	67.350	2.500	3.633
GCK	3.400	3.100	53.900	**47.450**	3.671	**2.738**
LEGENDRE	4.650	3.800	81.600	71.600	4.586	4.043
CHEBYSHEV	5.500	3.950	98.950	72.950	5.643	4.029
s-HERMITE	2.750	**3.050**	46.200	52.500	2.786	3.210
GEGENBAUER	**2.050**	3.450	**35.250**	51.150	**1.814**	3.348
p-value	**1.53E-09**	0.566	**0.005**	**0.004**	**1.60E-13**	0.028

5.2. Comparison of Gegenbauer versus classical kernels

The Gegenbauer kernel was compared against the linear, polynomial and RBF kernels. The results of this comparison are presented in Table 7. In terms of Accuracy, it can be observed that the Gegenbauer kernel achieved better performance than the linear and polynomial kernels. Taking the Gegenbauer kernel as a control method and carrying out *post hoc* tests it was confirmed that Gegenbauer is significantly better than the linear and polynomial kernels but not significantly different than the RBF kernel. In terms of PSV, Gegenbauer is not different than the linear and polynomial kernels, but it is better, with statistical significance, than the RBF kernel.

5.3. Comparison of Gegenbauer versus orthogonal polynomial kernels

Having compared the Gegenbauer kernel against classical kernels, now we compare Gegenbauer against the orthogonal polynomial kernels. The results, reported in Table 8, show that with respect to accuracy, Gegenbauer was ranked first in all the tests. *Post hoc* tests revealed that the Gegenbauer kernel is better than all orthogonal polynomial kernels except for our modified kernel s-Hermite and MCK. In terms of PSV, the *post hoc* tests show that Gegenbauer is better, with statistical significance, than the Chebyshev kernel only.

As a general conclusion, the Gegenbauer and the s-Hermite kernels are generally better in classification accuracy than the rest of the orthogonal polynomial kernels and maintain a low PSV. The RBF obtains the second best rank overall according to the accuracy index, but associated with the worst rank overall according to the PSV index. In accordance with results of previous studies [13,14], radial kernels in general require a larger amount of support vectors than the orthogonal polynomial kernels. The theory indicates that this may be detrimental to their generalization capabilities.

5.4. Hyper-parameter optimization results

The best values for the different parameters of the kernels discussed in this work are reported in Tables 9–11 and are the result of the optimization performed as part of the experimental methodology described above. The main focus in this work is not hyper-parameter optimization, however, Tables 9–11 contain important information that can be useful as a reference for those working in that problem, as well as to ensure the reproducibility of our study. The best parameter values were computed based on the results of the best 25% of the SVMs population, sorted by accuracy.

As can be appreciated, the values reported in these tables present a large variability and because of this, it becomes difficult to describe them in brief. Nevertheless, an important observation to be made is that a large standard deviation (relative to the range in which each parameter was optimized), indicates that the parameter has a weak influence in the performance of the kernel on the corresponding dataset. In the case of the polynomial degree in Table 11, a large proportion of occurrence of a particular degree indicates that the degree is the best choice of this parameter for the dataset, while a small proportion indicates that different degrees can produce very similar performance and the reported value corresponds to the majority of cases.

Table 9 contains the best values of the parameter C for each combination of a kernel and a dataset. This parameter is required by the SVM classifiers independently of the type of kernel function. As can be observed, for a significant number of cases, the standard deviation obtained is relatively large when compared with the range on which this parameter was optimized. This observation indicates that C has a lower impact on performance than other parameters, as will be shown next.

Table 10 shows the best value found for the intrinsic parameters of kernels, these are: γ (RBF and MCK), a (polynomial) and α (Gegenbauer). In this table, the reported standard deviations are relatively small, which unlike the case of the C parameter, indicates that the intrinsic kernel parameters possess a stronger influence on the performance of SVM classifiers.

Table 11 contains the best value of the polynomial degree n for the polynomial kernels (GCK, Legendre, Chebyshev, s-Hermite, MCK, Polynomial and Gegenbauer); the values appearing within parentheses represent the highest percentage of occurrence of a particular degree within the top 25% of the evaluated SVMs, sorted by accuracy and without duplicates. A relatively small value, such as (0.20) for MCK on dataset 'a1a', indicates large diversity in the parameters of the best solutions (i.e. other degrees are close in performance to the optimum); in contrast, a value close to 1, such as (0.96) for MCK on dataset 'Australian', indicates dominance of the corresponding degree within the best solutions found. Please notice that the results in Table 11 should not be interpreted in isolation from other results. For instance, the fact that the best degree found for the Polynomial kernel on dataset 'Monks-3' is $n = 1$ with a high proportion of occurrence (0.87), does not necessarily mean that said dataset is easily solved by an essentially linear kernel. This becomes apparent by looking at the classification accuracy of the Polynomial kernel (*cf.* Table 3), according to which the kernel is ranked second from last, for that same dataset.

Interestingly, in Table 11 the smaller degrees (1, 2 and 3) appear most often than the larger degrees (4, 5 and 6), both globally

Table 9
Best values of the C parameter, Mean ± Std. Dev.

Dataset	RBF	MCK	LINEAR	GCK	POLY	LEGEN	CHEBY	s-HERM	GEGEN
a1a	10.45 ± 3.7	15.18 ± 7.72	22.34 ± 9.7	8.66 ± 8.07	18.65 ± 7.5	15.12 ± 8.9	18.77 ± 7.9	12.21 ± 6.7	24.46 ± 3.6
Australian	14.25 ± 6.2	5.06 ± 5.03	10.88 ± 9.2	8.85 ± 8.15	14.44 ± 7.6	2.22 ± 3.6	18.84 ± 9.0	18.70 ± 6.8	24.28 ± 4.8
Breast	19.34 ± 1.1	15.27 ± 9.21	16.31 ± 12.2	5.19 ± 5.24	17.38 ± 8.1	16.09 ± 9.6	17.42 ± 9.3	7.44 ± 4.0	16.25 ± 8.3
Diabetes	11.83 ± 0.2	7.41 ± 6.43	6.70 ± 0.6	10.19 ± 8.26	13.92 ± 7.2	5.43 ± 3.1	3.91 ± 1.2	16.43 ± 7.5	18.75 ± 8.4
Fourclass	30.42 ± 1.5	15.72 ± 9.86	11.77 ± 4.3	18.98 ± 7.67	21.63 ± 7.7	31.58 ± 1.0	28.85 ± 1.9	25.20 ± 4.4	31.52 ± 0.7
German	13.12 ± 2.0	15.93 ± 8.84	18.67 ± 9.2	0.75 ± 0.69	13.46 ± 8.3	15.761 ± 0.0	13.51 ± 8.5	3.18 ± 3.1	16.89 ± 7.1
Glass2	23.49 ± 1.4	11.99 ± 9.73	0.29 ± 0.3	10.03 ± 9.38	15.46 ± 9.1	17.98 ± 8.2	16.65 ± 11.4	16.65 ± 9.9	11.07 ± 11.0
Haberman	12.52 ± 1.1	9.38 ± 7.14	0.79 ± 0.4	10.72 ± 8.53	20.51 ± 6.3	15.56 ± 9.5	1.85 ± 0.5	25.26 ± 3.6	21.13 ± 5.3
Heart	9.64 ± 0.1	12.68 ± 10.01	9.35 ± 1.0	6.23 ± 9.59	19.18 ± 4.9	14.24 ± 9.3	13.42 ± 7.6	9.65 ± 2.4	11.58 ± 6.9
Ionosphere	18.30 ± 8.2	14.33 ± 7.97	30.04 ± 1.5	9.51 ± 8.27	10.57 ± 5.1	19.31 ± 6.3	12.43 ± 4.0	10.77 ± 4.3	5.32 ± 3.7
Liver	29.08 ± 3.0	10.81 ± 7.05	20.37 ± 0.1	14.64 ± 8.52	12.55 ± 5.0	15.22 ± 6.4	22.11 ± 3.9	28.53 ± 3.1	25.30 ± 4.8
Monks-1	16.86 ± 0.0	14.61 ± 6.89	1.00 ± 1.0	11.80 ± 11.29	6.24 ± 1.0	15.94 ± 8.1	18.20 ± 8.4	20.61 ± 0.5	20.96 ± 3.4
Monks-2	21.64 ± 2.4	15.96 ± 7.01	0.21 ± 0.1	12.92 ± 7.76	12.70 ± 6.6	5.44 ± 4.4	17.59 ± 9.1	29.84 ± 2.0	18.06 ± 5.8
Monks-3	29.17 ± 1.4	13.15 ± 8.20	1.37 ± 0.2	1.62 ± 2.14	6.13 ± 1.8	0.05 ± 0.0	10.55 ± 6.9	29.42 ± 1.8	27.04 ± 5.0
plrx	23.63 ± 7.9	15.45 ± 9.86	21.51 ± 9.0	10.94 ± 10.07	14.99 ± 7.8	13.72 ± 9.1	21.43 ± 11.2	23.13 ± 8.0	12.68 ± 9.1
Sonar	21.93 ± 2.4	14.16 ± 7.46	0.42 ± 0.0	14.18 ± 8.54	20.43 ± 7.9	16.96 ± 8.6	19.58 ± 9.8	20.44 ± 8.6	17.60 ± 9.3
Splice	9.77 ± 1.2	14.47 ± 6.73	0.88 ± 0.7	9.64 ± 9.76	17.08 ± 7.8	14.27 ± 7.6	13.75 ± 10.1	19.95 ± 9.8	19.03 ± 7.9
Vehicle	28.55 ± 1.8	9.69 ± 6.27	17.15 ± 4.4	14.31 ± 9.48	12.58 ± 7.2	27.45 ± 6.1	18.75 ± 8.7	22.24 ± 6.0	20.20 ± 7.5
wdbc	22.39 ± 3.6	15.01 ± 8.47	10.04 ± 2.5	8.01 ± 8.00	14.68 ± 9.1	16.95 ± 9.2	22.03 ± 5.3	24.64 ± 5.4	20.23 ± 8.4
wpbc	27.84 ± 6.2	14.76 ± 8.74	6.59 ± 2.2	13.65 ± 9.36	17.23 ± 3.1	14.12 ± 10.3	15.02 ± 8.7	10.07 ± 5.7	20.48 ± 3.6

Table 10
Best value of intrinsic kernel parameter, Mean ± Std. Dev.

Dataset	RBF γ	MCK γ	POLY a	GEGEN α
a1a	0.02 ± 0.0	0.23 ± 0.08	2.06 ± 1.0	0.96 ± 0.2
Australian	0.06 ± 0.0	0.31 ± 0.35	0.30 ± 0.2	1.20 ± 0.2
Breast	0.03 ± 0.0	1.94 ± 0.76	1.55 ± 1.1	0.50 ± 0.3
Diabetes	0.18 ± 0.0	0.34 ± 0.45	1.71 ± 1.0	0.61 ± 0.3
Fourclass	3.82 ± 0.1	1.12 ± 0.68	3.33 ± 0.3	−0.42 ± 0.1
German	0.02 ± 0.0	0.41 ± 0.24	2.01 ± 1.0	0.85 ± 0.4
Glass2	3.13 ± 0.4	1.77 ± 1.29	0.58 ± 0.9	0.38 ± 0.3
Haberman	0.71 ± 0.0	0.54 ± 0.67	2.97 ± 0.5	0.00 ± 0.2
Heart	0.02 ± 0.0	0.24 ± 0.21	2.55 ± 0.7	0.84 ± 0.2
Ionosphere	0.05 ± 0.0	0.70 ± 0.50	0.14 ± 0.0	1.36 ± 0.1
Liver	0.59 ± 0.2	0.39 ± 0.44	1.44 ± 0.2	0.23 ± 0.1
Monks-1	0.08 ± 0.0	0.05 ± 0.05	0.34 ± 0.0	0.38 ± 0.0
Monks-2	0.43 ± 0.0	0.15 ± 0.27	2.39 ± 1.0	−0.44 ± 0.1
Monks-3	0.08 ± 0.0	0.17 ± 0.22	0.36 ± 0.2	1.11 ± 0.4
plrx	0.05 ± 0.0	3.20 ± 0.72	0.02 ± 0.0	0.51 ± 0.0
Sonar	0.04 ± 0.0	0.11 ± 0.07	1.69 ± 1.1	0.74 ± 0.1
Splice	0.02 ± 0.0	0.04 ± 0.02	2.25 ± 1.2	0.74 ± 0.1
Vehicle	1.27 ± 0.0	0.16 ± 0.13	1.32 ± 0.3	0.76 ± 0.4
wdbc	0.12 ± 0.0	0.41 ± 0.26	0.62 ± 0.5	0.61 ± 0.2
wpbc	0.07 ± 0.0	0.75 ± 0.53	2.71 ± 0.5	0.83 ± 0.1

and among the results with high proportion of occurrence (≥ 0.5). This indicates that, in general, the best results of polynomial-based kernels are achieved by simpler functions, rather than more complex ones. This observation can be counterintuitive and we consider that it justifies an in-depth analysis of the optimal parameters of SVM kernels, which unfortunately is beyond the scope of the present study.

The performance distribution of all Gegenbauer-based classifiers, in terms of accuracy (black crosses) and PSV (red points) is illustrated in Fig. 6 for every dataset; darker shades indicate better classifiers. Notice that the plot for each dataset contains about 52,500 pairs of points, each pair corresponding to an SVM classifier that was trained and evaluated. Altogether the plots contain the performance indexes of more than 1,000,000 classifiers as function of the parameter α. This plot is a useful tool for the analysis of the SVMs behavior with Gegenbauer kernels.

A high accuracy and a small PSV is desirable of any SVM classifier. This combination was found most often for $\alpha \in [0.3, 0.9]$. In the plots, the range of α that produces the best classifiers is indicated by a shaded rectangle. SVM classifiers can suffer from overfitting (PSV 70% or higher); this can be seen in the case of datasets like 'Monks-1', 'Monks-3', and 'Liver' for $\alpha > 0.5$ and the

Table 11
Best values of the polynomial degree (proportion of occurrence within top 25%).

Dataset	MCK	GCK	POLY	LEGEN	CHEBY	s-HERM	GEGEN
a1a	6 (0.20)	1 (0.64)	1 (1.00)	2 (0.85)	4 (0.38)	4 (0.37)	2 (0.19)
Australian	1 (0.96)	1 (0.87)	2 (0.55)	6 (0.31)	1 (0.67)	5 (0.28)	5 (0.24)
Breast	1 (0.31)	1 (0.76)	1 (0.47)	4 (0.44)	3 (0.60)	4 (0.25)	5 (0.21)
Diabetes	1 (0.52)	1 (0.50)	2 (0.48)	2 (0.91)	1 (0.82)	6 (0.28)	2 (0.23)
Fourclass	4 (0.19)	5 (0.25)	5 (1.00)	5 (0.39)	3 (0.26)	6 (0.47)	6 (0.36)
German	1 (0.25)	1 (0.90)	1 (0.38)	3 (0.94)	4 (0.43)	5 (0.30)	1 (0.19)
Glass2	1 (0.28)	4 (0.41)	2 (0.25)	2 (0.44)	6 (0.30)	1 (0.50)	2 (0.24)
Haberman	1 (0.36)	2 (0.45)	4 (0.49)	3 (0.58)	1 (0.85)	6 (0.50)	2 (0.21)
Heart	1 (0.26)	1 (0.43)	1 (0.33)	3 (0.67)	1 (0.69)	6 (0.28)	2 (0.19)
Ionosphere	1 (0.53)	1 (0.70)	2 (0.83)	3 (0.55)	3 (0.48)	5 (0.35)	5 (0.21)
Liver	2 (0.38)	2 (0.39)	2 (0.40)	2 (0.48)	1 (0.75)	2 (0.27)	3 (0.23)
Monks-1	4 (0.25)	1 (0.46)	2 (0.58)	3 (0.53)	3 (0.61)	2 (0.49)	2 (0.43)
Monks-2	2 (0.63)	1 (0.80)	2 (0.98)	2 (1.00)	3 (0.57)	2 (0.54)	1 (0.89)
Monks-3	1 (0.28)	1 (0.62)	1 (0.87)	3 (0.62)	1 (0.46)	6 (0.25)	2 (0.23)
plrx	1 (0.29)	2 (0.43)	2 (0.30)	6 (0.33)	4 (0.27)	1 (0.19)	4 (0.20)
Sonar	1 (0.23)	1 (0.45)	3 (0.33)	2 (1.00)	1 (1.00)	2 (0.32)	5 (0.25)
Splice	1 (0.48)	1 (0.72)	3 (0.46)	2 (0.78)	1 (0.60)	5 (0.66)	2 (0.21)
Vehicle	1 (0.84)	1 (0.67)	3 (0.51)	2 (0.96)	1 (0.50)	6 (0.44)	5 (0.21)
wdbc	1 (0.24)	1 (0.71)	1 (0.43)	2 (0.56)	1 (0.65)	4 (0.25)	2 (0.21)
wpbc	1 (0.38)	1 (0.68)	1 (0.84)	3 (0.43)	3 (0.60)	2 (0.50)	1 (0.37)

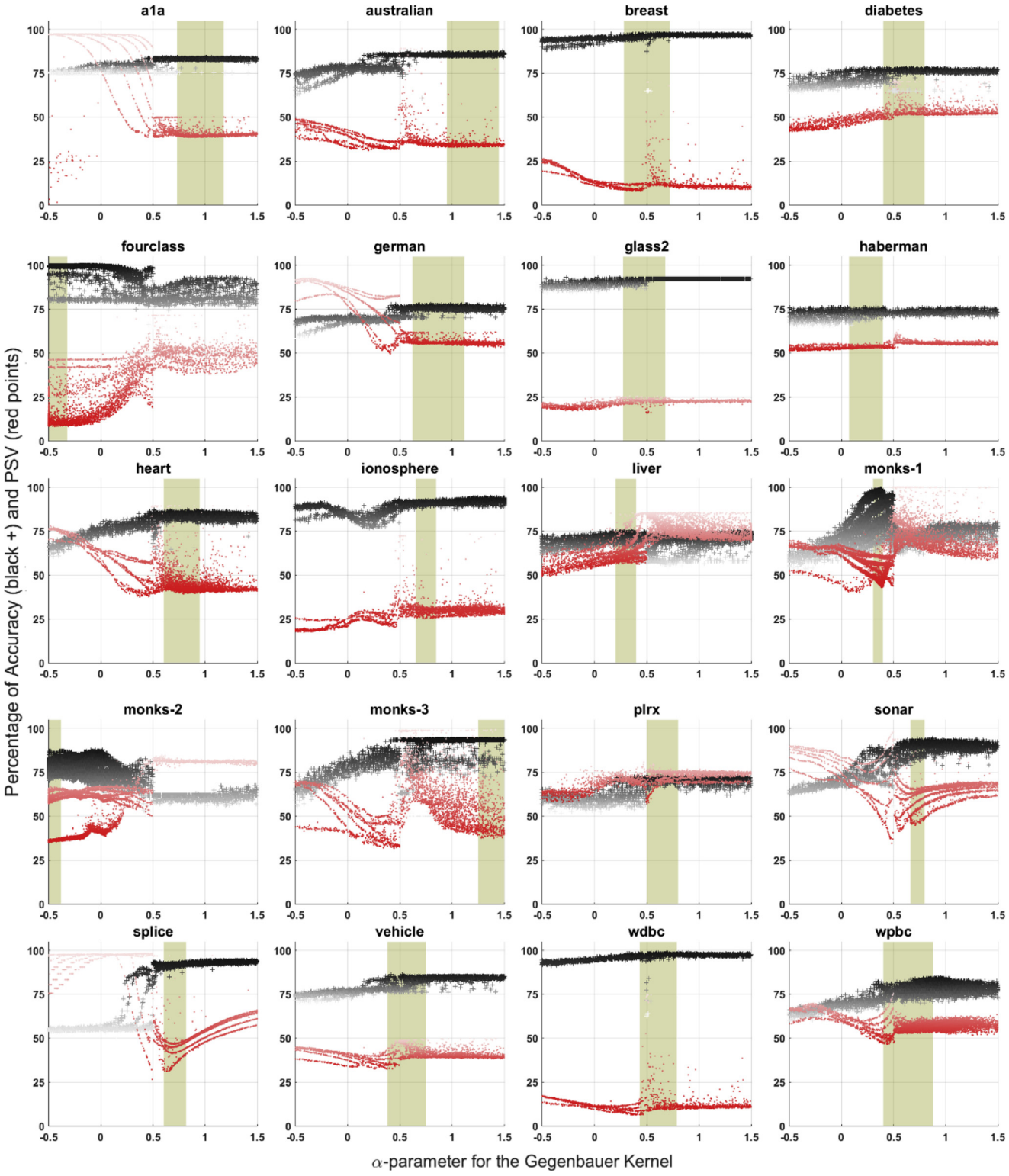

Fig. 6. Performance of the Gegenbauer spectrum of classifiers obtained by varying its α parameter. Red dots represent the PSV while black '+' symbols represent the Accuracy. Darker shades indicate classifiers with higher performance metrics. The ranges in which the optimal values of the α-parameter were found are indicated by a shaded rectangle in the corresponding plot.

dataset 'Splice' for $\alpha < 0.5$. For the majority of the datasets, the PSV values exhibit a characteristic behavior: points belonging to the same polynomial degree are grouped together and form distinctive curves which are distributed along the vertical direction. For the accuracy values, this behavior is much less notorious with the exception of a few datasets like 'Fourclass' and 'Splice'.

Summary of findings and observations
Gegenbauer and s-Hermite:

- Obtained better combinations of accuracy and PSV indexes than the rest of the kernels.

- The hypothesis that these kernels require a smaller PSV than other kernels and maintain high accuracy is confirmed.
- Polynomials of high degrees (e.g. bigger than 6) have higher computational demand compared to lower degrees; our proposed kernels were able to outperform the rest of the polynomial kernels using only small degrees.

Findings related to the parameter α in the Gegenbauer family:

- $\alpha = 0$, special case (Chebyshev type-I), obtained the worst results within the family.
- $0.3 \leq \alpha \leq 0.9$, most of the best Gegenbauer-based classifiers were found in this interval.
- $\alpha > 0.5$, classifiers in this interval are fairly equivalent to each other in terms of accuracy. PSV becomes a valuable criterion to define the optimum.
- $\alpha < 0.5$, the polynomials in this interval already have an appropriate amplitude and therefore a scaling and weight functions are unnecessary to avoid the Explosion and Annihilation effects.

6. Conclusions and future work

The most important conclusion is that orthogonal polynomial kernels following our proposed formulation are as effective as the benchmark RBF kernel, with the added advantage of being more efficient with respect to the required amount of support vectors. It is our understanding that this behavior is due to two main reasons: first, that the orthogonal polynomials reduce the data representation redundancy in the feature space, leading to less data points being considered as support vectors; second, that our formulation allows the generation of a large amount of new kernels from which an optimum classifier may be obtained.

Another conclusion is that the Gegenbauer kernel family includes Legendre and Chebyshev kernels, with improved performance. Within the Gegenbauer family, many other classifiers were found that overcome these two special cases. Under our formulation, Gegenbauer should be preferred over other orthogonal polynomials because the required scaling function can be analytically computed (contrary, for instance, to Hermite).

Finally, our results suggest that a trade-off between accuracy and PSV indexes must always be considered when working with SVM classifiers, since employing just one of these hides relevant information that can affect the generalization performance of a classifier. Our future work includes exploring the optimization of classifiers based on a combined performance index and further improvements to our orthogonal polynomial kernel formulation.

Acknowledgment

This work was partially supported by the National Council of Science and Technology of Mexico (CONACYT) through grants: 375524 (Luis C. Padierna) and CATEDRAS-2598 (A. Rojas).

References

[1] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, 2004.
[2] N. Cristianini, J. Shawe-Taylor, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, 2004.
[3] A. Shigeo, Support Vector Machines for Pattern Classification, Springer, New York, 2010.
[4] N. Deng, Y. Tian, C. Zhang, Support Vector Machines, CRC Press, Boca Ratón, 2013.
[5] J. Shawe-Taylor, S. Sun, A review of optimization methodologies in support vector machines, Neurocomputing (2011) 3609–3618.
[6] V. Vapnik, Statistical Learning Theory, New York, John Wiley and Sons, 1998.
[7] C. Hsu, C. Chang, C. Lin, A practical guide to support vector classification, 2003. [Online]. Available http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
[8] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, J. Mach. Learn. Res. 2 (2002) 419–444.
[9] A.D. Essam, T. Hamza, New empirical nonparametric kernels for support vector machines classification, Appl. Soft Comput. (13) (2013) 1759–1765.
[10] N. Ye, R. Sun, Y. Liu, L. Cao, Support vector machine with orthogonal Chebyshev kernel, 18th International Conference on Pattern Recognition, Washington, 2006.
[11] Z. Pan, H. Chen, X. You, Support vector machine with orthogonal Legendre kernel, International Conference on Wavelet Analysis and Pattern Recognition, Xian, 2012.
[12] S. Ozer, C.H. Chen, Generalized Chebyshev kernels for support vector classification, ICPR International Conference On Pattern Recognition, Tampa, FL, USA, 2008.
[13] S. Ozer, C. Chen, H. Cirpan, A set of new Chebyshev kernel functions for support vector machine pattern classification, Pattern Recognit. 44 (7) (2011) 1435–1447.
[14] H.V. Moghaddam, J. Hamidzadeh, New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier, Pattern Recognit. 60 (2016) 921–935.
[15] L. Sun, K.-A. Toh, Z. Lin, A center sliding Bayesian binary classifier adopting orthogonal polynomials, Pattern Recognit. 48 (6) (2015) 2013–2028.
[16] Y. Liu, S. Liao, H. Lin, Y. Yue, W. Wang, Infinite Kernel learning: generalization bounds and algorithms, Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), 2017.
[17] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1–3) (2002) 131–159.
[18] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, in: Philosophical Transactions of the Royal Society of London, Series A, 1909, pp. 415–446.
[19] W. Gautschi, Orthogonal polynomials: applications and computation, Acta Numer. (1996) 45–119.
[20] M. Parodi, J.C. Gómez, Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations, Pattern Recognit. 47 (1) (2014) 128–140.
[21] J. Zhao, G. Yan, B. Feng, W. Mao, J. Bai, An adaptive support vector regression based on a new sequence of unified orthogonal polynomials, Pattern Recognit. (2013) 899–913.
[22] V. Totik, Orthogonal polynomials, Surv. Approximation Theory (2005) 70–125.
[23] C. Dunkl, Y. Xu, Orthogonal Polynomials of Several Variables, second ed., Cambridge University Press, Cambridge, 2014.
[24] T. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach, New York, 1978.
[25] Y. Xu, On discrete orthogonal polynomials of several variables, Adv. Appl. Math. 33 (2004) 615–632.
[26] F. Olver, D. Lozier, R. Boisvert, C. Clark, NIST Handbook of Mathematical Functions, Cambridge University Press, New York, 2010.
[27] M. Tian, W. Wang, Some sets of orthogonal polynomial Kernel functions, Appl. Soft Comput. 61 (2017) 742–756.
[28] D.S. Kim, T. Kim, S.-H. Rim, Some identities involving Gegenbauer polynomials, Adv. Diff. Equat. 2012 (1) (2012) 219.
[29] B. Spencer, The Classical Orthogonal Polynomials (2015).
[30] N. Christianini, J. Shawe-Taylor, An Introduction to SVM and Other Kernel Based Methods., Cambridge University Press, Cambridge, U.K, 2000.
[31] S.I. Valdez, A. Hernández, S. Botello, A Boltzmann based estimation of distribution algorithm, Inf. Sci. (2013) 126–137.
[32] T. Meng, W. Wenjian, Some sets of orthogonal polynomial kernel functions, Appl. Soft Comput. 61 (2017) 742–756.
[33] A. López, X. Li, W. Yu, Support vector machine classification for large datasets using decision tree and fisher linear discriminant, Future Gener. Comput. Syst. 36 (2014) 57–65.
[34] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[35] J. Zhao, Z. Yang, X. Yitian, Nonparallel least square support vector machine for classification, Appl. Intell. (2016) 1–10.
[36] M. Tanveer, Newton method for implicit Lagrangian twin support vector machines, Int. J. Mach. Learn. Cybern. 6 (6) (2015) 1029–1040.
[37] M. Nekkaa, D. Boughaci, A memetic algorithm with support vector machine for feature selection and classification, Memetic Comput. 7 (1) (2015) 59–73.
[38] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.
[39] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.
[40] N. Verbiest, K. Vermeulen, A. Teredesai, Evaluation of classification methods, in: Data Classification: Algorithms and Applications, CRC Press, Boca Raton, 2015, pp. 633–652.

Luis Carlos Padierna received his Ph.D. in Computer Science from the Instituto Tecnológico de León in 2018. His research interests include support vector machines, machine learning, big data and their applications to industry and biomedicine.

Martín Carpio received his Ph.D. from the Universidad de Guanajuato, México, in 1995. Currently, he is a Senior Lecturer and Researcher at the Instituto Tecnológico de León. His research interests include computational optimization, simulation processes, statistical data analysis and evolutionary computation.

Alfonso Rojas-Domínguez received his Ph.D. from the University of Liverpool, UK, in 2007. Currently, he holds a position as a CONACYT Research Fellow at the Instituto Tecnológico de León. His research interests include automated image processing and analysis, computational intelligence and machine learning.

Héctor Puga received his Ph.D. from the Universidad de Guanajuato, México, in 2002. Currently, he leads the Intelligent Systems research group at the Instituto Tecnológico de León. His research interests include optimization, hyper-heuristics and intelligent systems.

Héctor Fraire received his Ph.D. in Computer Science from the Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México in 2005. His scientific interests include metaheuristic optimization and machine learning.